PB85-116069

GC/MS (Gas Chromatograph/Mass Spectrometer) Data Management Process Part 2. Program Documentation. Version 1.0

Cincinnati Univ. Computing Center, OH

Prepared for

Environmental Monitoring and Support Lab.
Cincinnati, OH

Oct 84

U.S. Department of Commerce
National Technical Information Service
NTIS

PB85-116069

EPA-600/4-84-084
October 1984

A GC/MS DATA MANAGEMENT PROCESS

Part II

Program Documentation

Version 1.0

by

Dennis P. Ryan
University of Cincinnati Computing Center
Cincinnati, Ohio 45221

and

John P. Donnelly
Physical and Chemical Methods Branch
Environmental Monitoring and Support Laboratory
Cincinnati, Ohio 45268

Co-Operative Agreement
CR 808555-02-0

ENVIRONMENTAL MONITORING AND SUPPORT LABORATORY
OFFICE OF RESEARCH AND DEVELOPMENT
U. S. ENVIRONMENTAL PROTECTION AGENCY
CINCINNATI, OHIO 45268

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U.S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

TECHNICAL REPORT DATA
(Please read Instructions on the reverse before completing)

1. REPORT NO.
EPA-600/4-84-084

2.

3. RECIPIENT'S ACCESSION NO.
PB8 5 1 16069

4. TITLE AND SUBTITLE
A GC/MS DATA MANAGEMENT PROCESS
PART II, PROGRAM DOCUMENTATION
VERSION 1.0

5. REPORT DATE
October 1984

6. PERFORMING ORGANIZATION CODE

7. AUTHOR(S)
Dennis P. Ryan and John P. Donnelly

8. PERFORMING ORGANIZATION REPORT NO.

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Environmental Monitoring and Support Laboratory
26 W. St. Clair Street
Cincinnati, Ohio 45268

10. PROGRAM ELEMENT NO.
IBD884

11. CONTRACT/GRANT NO.
CR 808555-02-0

12. SPONSORING AGENCY NAME AND ADDRESS
Environmental Monitoring and Support Laboratory
26 W. St. Clair Street
Cincinnati, Ohio 45268

13. TYPE OF REPORT AND PERIOD COVERED

14. SPONSORING AGENCY CODE
EPA/600/06

15. SUPPLEMENTARY NOTES

16. ABSTRACT

The GC/MS Data Management Process (DMP) is an example process for reducing the measurement and identification data produced by a GC/MS to that necessary for input to the EPA sample file control system. This example is structured for a Finnigan Incos Data System.

The DMP consists of sixteen programs written in Fortran IV for Data General minicomputers. Each module is described in a section of the Program Documentation (Part II) along with flow charts, output sample formats, and variable description lists. The Fortran source code is attached.

A User's Guide (Part I) for this process is also available.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS	b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group
Mass Spectroscopy		1402
Spectroscopic Analysis		0703
Organic Compounds		

18. DISTRIBUTION STATEMENT
Release to Public

19. SECURITY CLASS *(This Report)*
Unclassified

20. SECURITY CLASS *(This page)*
Unclassified

21. NO. OF PAGES
175

22. PRICE

EPA Form 2220-1 (Rev. 4-77) PREVIOUS EDITION IS OBSOLETE

NOTICE

This document has been reviewed in accordance with U.S. Environmental Protection Agency policy and approved for publication. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

FOREWORD

Environmental measurements are required to determine the quality of ambient waters and the character of waste effluents. The Environmental Monitoring and Support Laboratory - Cincinnati, conducts research to:

- Develop and evaluate methods to measure the presence and concentration of physical, chemical, and radiological pollutants in water, wastewater, bottom sediments, and solid waste.
- Investigate methods for the concentration, recovery, and identification of viruses, bacteria and other microbiological organisms in water; and to determine the responses of aquatic organisms to water quality.
- Develop and operate an Agency-wide quality assurance program to assure standardization and quality control of systems for monitoring water and wastewater.
- Develop and operate a computerized system for instrument automation leading to improved data collection, analysis, and quality control.

This document and the software described were developed in the Advanced Instrumentation Section of the Environmental Monitoring and Support Laboratory (EMSL) for an example process for reducing measurement and identification information from a gas chromatograph/mass spectrometer data system. This data reduction system packages production data for the EPA sample file control system.

Robert L. Booth

Robert L. Booth, Director
Environmental Monitoring and Support
Laboratory - Cincinnati

ABSTRACT

This part of the GC/MS Data Management Process (DMP) documentation provides an explanation of the software which comprises this package. The DMP is an example process for reducing the measurement and identification data produced by a GC/MS to that necessary for input to the EPA sample file control system. This example is structured for a Finnigan Incos Data System.

The DMP consists of sixteen programs written in Fortran IV for Data General minicomputers. Each module is described in a section of the document along with flow charts, output sample formats, and variable description lists. The Fortran source code is attached.

A User's Guide (Part I) for this process is also available.

CONTENTS

Foreword iii
Abstract iv
Figures vi
Tables vii

1. Introduction 1
2. DMPN - The QUAN report transfer program 4
3. DMPL - The sample log file modification program 7
4. DMPC - The Data Management Process command router program 11
5. DMPQ - The Quality Control/Method Template editor programs 14
   5.1 DMPQ1 - Skeleton QC/MT creator program 14
   5.2 DMPQ2 - The QC/MT editor and report generator program 15
   5.3 DMPQ3 - The QC/MT limit update program 20
6. DMPG - The data consolidator and report generator programs 23
   6.1 DMPG1 - The reduced data file generator 23
   6.2 DMPG2 - The measurement and QC assessment reporter program 25
7. DMPE - The reduced data file editor program 34
8. DMPR - The replicate statistics computation and report generation programs 35
   8.1 DMPR1 - The mode selection and file identification program 35
   8.2 DMPR2 - The replicate statistics computation program 36
   8.3 DMPR3 - The statistics reporting program 41
   8.4 DMPR4 - The QC/MT statistical update program 44
9. DMPP - The SFC run results production programs 47
   9.1 DMPP1 - The run results file generation program 47
   9.2 DMPP2 - The run results file transmission program 49
10. DMPU - The DMP utility program 58
11. DMP data and report files 59
   11.1 $.LG - The sample log file 59
   11.2 $.QN - The sample quantitation file 59
   11.3 $.QA - The sample qualitative identification file 61
   11.4 *.QC - The Quality Control/Method Template file 61
   11.5 $.RD - The reduced sample data file 65
   11.6 DMPRPTG.99 - The measurement and QC assessment report 65
   11.7 DMPRPTQ.99 - The QC/MT contents report 65
   11.8 DMPRPTR.99 - The replicate statistics report 68
   11.9 The SFC compatible run results file and the SFC-supplied quality control limit file 68

FIGURES

Number Page

1.1 Primary Data Management Functions.............................. 3
2.1 Flow Chart of the DMPN Program................................ 5
3.1 Flow Chart of the DMPL Program................................ 9
4.1 Flow Chart of the DMPC Program................................ 13
5.1 Flow Control of the DMPQ1 Program.............................. 16
5.2 Flow Chart of the DMPQ2 Program............................... 21
6.1 Flow Chart of the DMPG1 Program............................... 26
6.2 Flow Chart of the DMPG2 Program............................... 33
8.1 Flow Chart of the DMPR1 Program............................... 37
8.2 Flow Chart of the DMPR2 Program............................... 42
8.3 Flow Chart of the DMPR3 Program............................... 45
9.1a Flow Chart of the DMPP1 Program............................... 50
9.1b Flow Chart of the DMPP1 Program............................... 51
9.2 Flow Chart of the DMPP2 Program............................... 57
11.1a Log File Header Record Format................................. 60
11.1b Log File Data Management and Volumetric Parameters............. 60
11.2 Example Quan Report Produced by INCOS System.................. 62
11.3 Record Layout for Qualitative Results......................... 63
11.6 Output Format of Various Measurement and Quality Control Report Records... 67
11.7 Format of the Quality Control Method Template Report........... 69
11.8 Header and Various Record Layouts for the Replicate Statistics Report.. 70

TABLES

Number		Page
2.1	Variables Descriptions for the DMPN Program	4
3.1	Variables Descriptions for the DMPL Program	10
4.1	Variables Descriptions for the DMPC Program	12
5.1	Variables Descriptions for the DMPQ Programs	17
6.1	Variables Descriptions for the DMPG Programs	27
8.1	Variables Descriptions for the DMPR Programs	38
9.1	Variables Descriptions for the DMPP Programs	52
11.4	GC/Method Template Description	64
11.5a	General Header Information in the Processed Data File	66
11.5b	Fixed List Quantification Record for the Processed Data File	66
11.9a	Type "1" SFC Record	71
11.9b	Type "2" SFC Record	72
11.9c	Type "4" SFC Record	73
11.9d	Format of the SFC Run Results Sample Header Record (Type "6")	74
11.9e	Format of the SFC Run Results Free Text Record (Type "7")	75

SECTION 1
INTRODUCTION

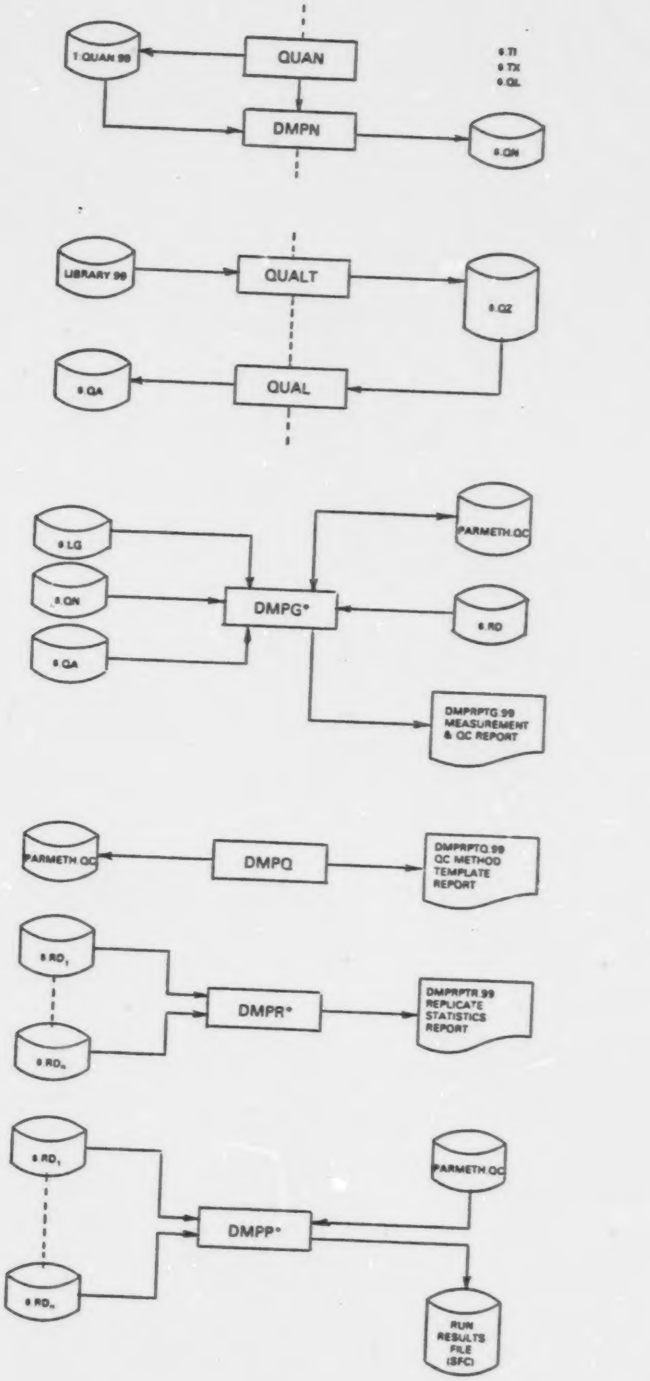
The Data Management Process (DMP) provides enhancements to the Finnigan INCOS GC/MS data system consisting of 16 computer programs which perform various computational and report oriented functions on reduced sample data. The data management programs are written in the FORTRAN IV language for the Data General NOVA 3 minicomputer, and they execute through MSDS. This document will describe the flow and interrelationship of each of the programs in the DMP subsystem. It also describes the data and report files maintained and generated by DMP.

The DMP subsystem will:

- gather quantitative and identification information on samples;
- generate reduced data files on each sample;
- produce measurement and quality control reports on sample data;
- compute replicate statistics for various parameters, using reduced data files; and
- package sample data in a compatible format for transmission to Sample File Control (SFC) an EPA data base management system

An integral part of the DMP subsystem is the Quality Control/Method Template (QC/MT) file. It is created and maintained by the system and provides quality control assessment criteria to various programs in the DMP subsystem.

The primary DMP functions and associated data and report files are shown in Figure 1.1. Each of these and the auxiliary functions are be described in a Section (Section 2-10) of this document. Each section contains a narrative description of the module (which may contain one or more programs), a flow chart of the module, a definition table for all module variables, and the appropriate program listings. Section 11 describes all data and report files manipulated by the DMP system.


1 QUAN 99
QUAN
DMPN
9.TI
9.TX
9.QL
9.QN
LIBRARY 99
QUALT
9.QZ
QUAL
9.QA
9.LG
9.QN
9.QA
DMPG*
PARMETH.QC
9.RD
DMPRPTG.99 MEASUREMENT & QC REPORT
PARMETH.QC
DMPQ
DMPRPTQ.99 QC METHOD TEMPLATE REPORT
9.RD₁
9.RDₙ
DMPR*
DMPRPTR.99 REPLICATE STATISTICS REPORT
9.RD₁
9.RDₙ
DMPP*
PARMETH.QC
RUN RESULTS FILE (SFC)

a) Creating QUAN files

b) Creating QUAL files

c) Gathering data and generating QC and measurement reports

d) Editing QC method template and generating contents

e) Generating replicate statistics report

f) Generating SFC run results files

*NameList may be used for batch process

Figure 1.1 Primary Data Management Functions

SECTION 2

2. DMPN - THE QUAN REPORT TRANSFER PROGRAM

The DMPN program is designed to execute in a user procedure just after the QUAN process has been completed. The DMPN program copies the extant QUAN.99 report file to the sample quantitation file ($.QN) and then deletes the QUAN.99 file. (Section 11.2 describes this file.) If QUAN.99 does not exist, an error message is displayed. A delay loop allows the analyst time to read the message before the procedure continues.

Figure 2.1 shows the flow of the DMPN program and Table 2.1 identifies the variables used in the program.

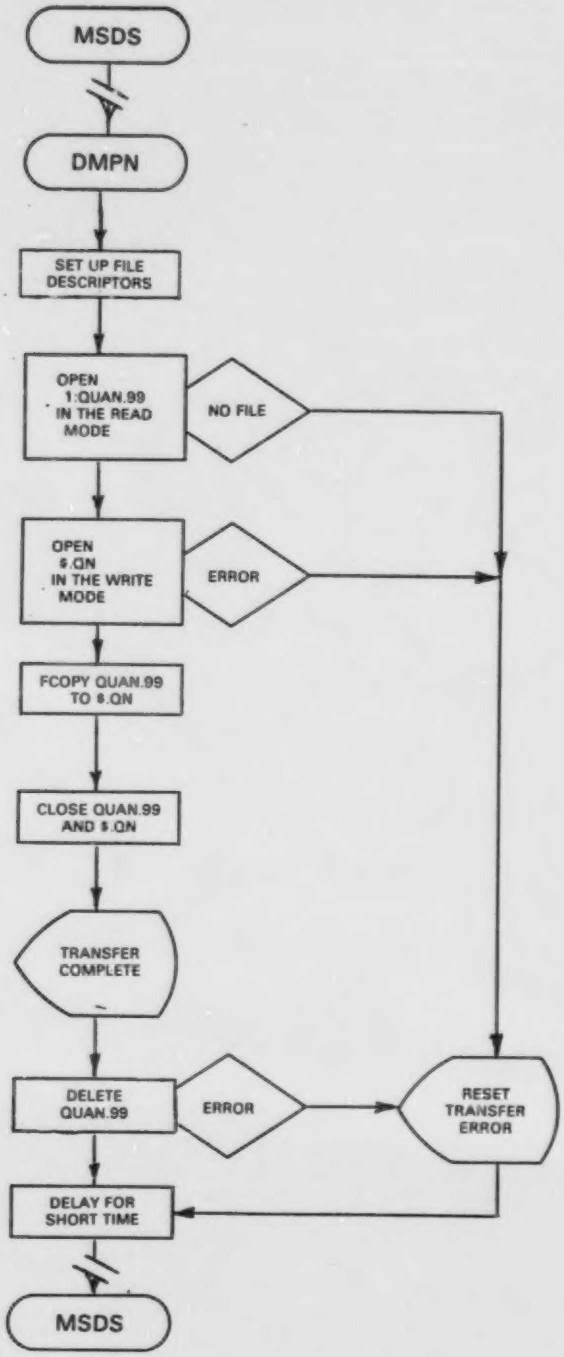

MSDS
DMPN
SET UP FILE DESCRIPTORS
OPEN 1:QUAN.99 IN THE READ MODE
NO FILE
OPEN $.QN IN THE WRITE MODE
ERROR
FCOPY QUAN.99 TO $.QN
CLOSE QUAN.99 AND $.QN
TRANSFER COMPLETE
DELETE QUAN.99
ERROR
RESET TRANSFER ERROR
DELAY FOR SHORT TIME
MSDS

Figure 2.1 Flow Chart of the DMPN Program

TABLE 2.1

VARIABLES DESCRIPTIONS FOR THE DMPN PROGRAM

VARIABLE	DESCRIPTION
ICHAN	disk channel number
IDESC(11)	sample descriptor
IE/IER	file I/O error codes
IQNFN(10)	quantitation file specifier
ISFILE(*,1)	MSDS data file descriptor (%)
JDESC(11)	1:QUAN.99 file name in descriptor form
KCHAN	channel number
LQN1	length variable
LQN2	length variable
LQN3	length variable
TTO	terminal I/O channel number
X	delay loop variable

3. DMPL - THE SAMPLE LOG FILE MODIFICATION PROGRAM

The DMPL program facilitates the user-entry of sample parameters into the sample log file ($.LG) which was created by INCOS. (The sample log file is described in section 11.1.) The program is designed to execute from MSDS or from a user procedure immediately following data acquisition.

The DMPL program first gets the date and time from the system clock. It then gets the current sample name from the ISFILE array and prompts the user to accept the name or input a new sample name. Once the program certifies the presence of the log file, it opens the file in the read mode and reads it in search of the '$$$' string. If it finds the '$$$' string, it presumes that the parameters were previously entered. It reads the parameters and shifts control to the user entry segment of the program. If no '$$$' string is encountered, then the parameters are all set to default values. The log file is closed in either case. The user entry segment prompts for each parameter. When the session is complete, the program prompts for the analysts acceptance or rejection of the parameters. Once an acceptable set of parameters has been obtained, the program opens the log file in the modify mode. If the file was previously modified, the program searches for the '$$$' string and writes the updated parameters in the appropriate fields. If the parameters are new, the program positions the file pointer to the end of the log file, writes out the date and time, writes out the

'$$$' string, and writes out the new parameters in the appropriate fields. Once DMPL is finished modifying the log file, it closes the file and chains the executive.

Figure 3.1 shows the flow of the DMPL program and Table 3.1 identifies the variables used in the program.

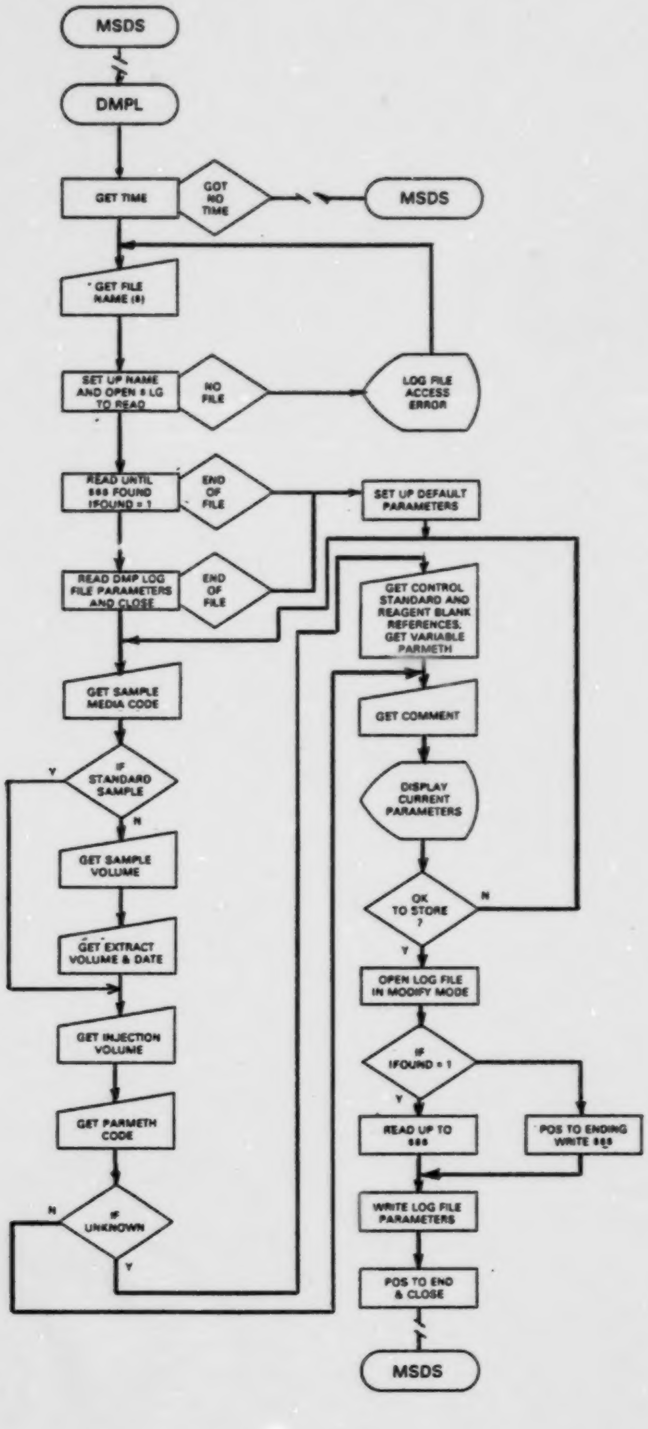

MSDS
DMPL
GET TIME
GOT NO TIME
MSDS
* GET FILE NAME (8)
SET UP NAME AND OPEN $ LG TO READ
NO FILE
LOG FILE ACCESS ERROR
READ UNTIL $$$ FOUND IFOUND = 1
END OF FILE
SET UP DEFAULT PARAMETERS
READ DMP LOG FILE PARAMETERS AND CLOSE
END OF FILE
GET CONTROL STANDARD AND REAGENT BLANK REFERENCES. GET VARIABLE PARMETH
GET SAMPLE MEDIA CODE
GET COMMENT
IF STANDARD SAMPLE
Y
N
DISPLAY CURRENT PARAMETERS
GET SAMPLE VOLUME
OK TO STORE ?
N
Y
GET EXTRACT VOLUME & DATE
OPEN LOG FILE IN MODIFY MODE
GET INJECTION VOLUME
IF IFOUND = 1
Y
GET PARMETH CODE
READ UP TO $$$
POS TO ENDING WRITE $$$
IF UNKNOWN
N
Y
WRITE LOG FILE PARAMETERS
POS TO END & CLOSE
MSDS

Figure 3.1 Flow Chart of the DMPL Program

TABLE 3.1

VARIABLES DESCRIPTIONS FOR THE DMPL PROGRAM

VARIABLE	DESCRIPTION
IANS(5)	answer string
ICH	channel number
ICMNT(32)	user comment string
ICRDF	duplicate/spike sample type character
ICTST(5)	control standard reference name
IDAY	day
IE/IER	I/O error codes
IEXDT(4)	sample extraction date
IEXTL(2)	log filename extension
IFLFN(10)	filename specifier
IFLNM(8)	filename without extension
IFOUND	flag indicating presence of '$$$' string
IHR	hour
IMIN	minute
IMO	month
IP1	string position variable
IPMTH(4)	PARMETH string for fixed list results
IPOS	string position variable
IPVTH(4)	PARMETH string for variable list results
IRGBK(5)	reagent blank reference
ISEC	seconds
ISOURCE	sample media indicator
ITIME(9)	time/date string
ITMP(20)	text string
ITXT(50)	text string
IVS(2)	sample media identification string
IX	dummy integer
IYR	year
LN	length of sample name
LNN	LN+1
LOOK2(2)	'$$$' string
MATCH(8)	sample media super string
MCODE(4)	sample type super string
VEX	extract volume
VINJ	injection volume
VOL	sample volume
XTMP	default real value

SECTION 4

4. DMPC - THE DATA MANAGEMENT PROCESS COMMAND ROUTER PROGRAM

The DMPC Program is capable of routing control to other programs which perform the major DMP subsystem activities. Any of six functions may be accessed from DMP command level with additional features provided for HELP and exit to MSDS.

The DMPC program first gets the time from the system clock. If time is unavailable, control transfers to MSDS. Once time has been obtained, the user is prompted for the QC/MT file name to be used in subsequent operations. The name is checked for validity and then the DMPC program presents the command line options to the analyst. If HELP is selected, the program provides a display of the options and a description of each. After the user enters a command symbol, the program searches the command string for the presence of the symbol. If it is valid, the program checks the option enable flag to determine its availability. The DMPC program then chains to the appropriate program.

The DMPC program establishes one small area common to all the rest of the programs described in this booklet. This area holds the QC/MT name (IQCNM(4)) and the process date (IPRDT(4)).

Figure 4.1 shows the flow of the DMPC program and Table 4.1 identifies the variables used in the program.

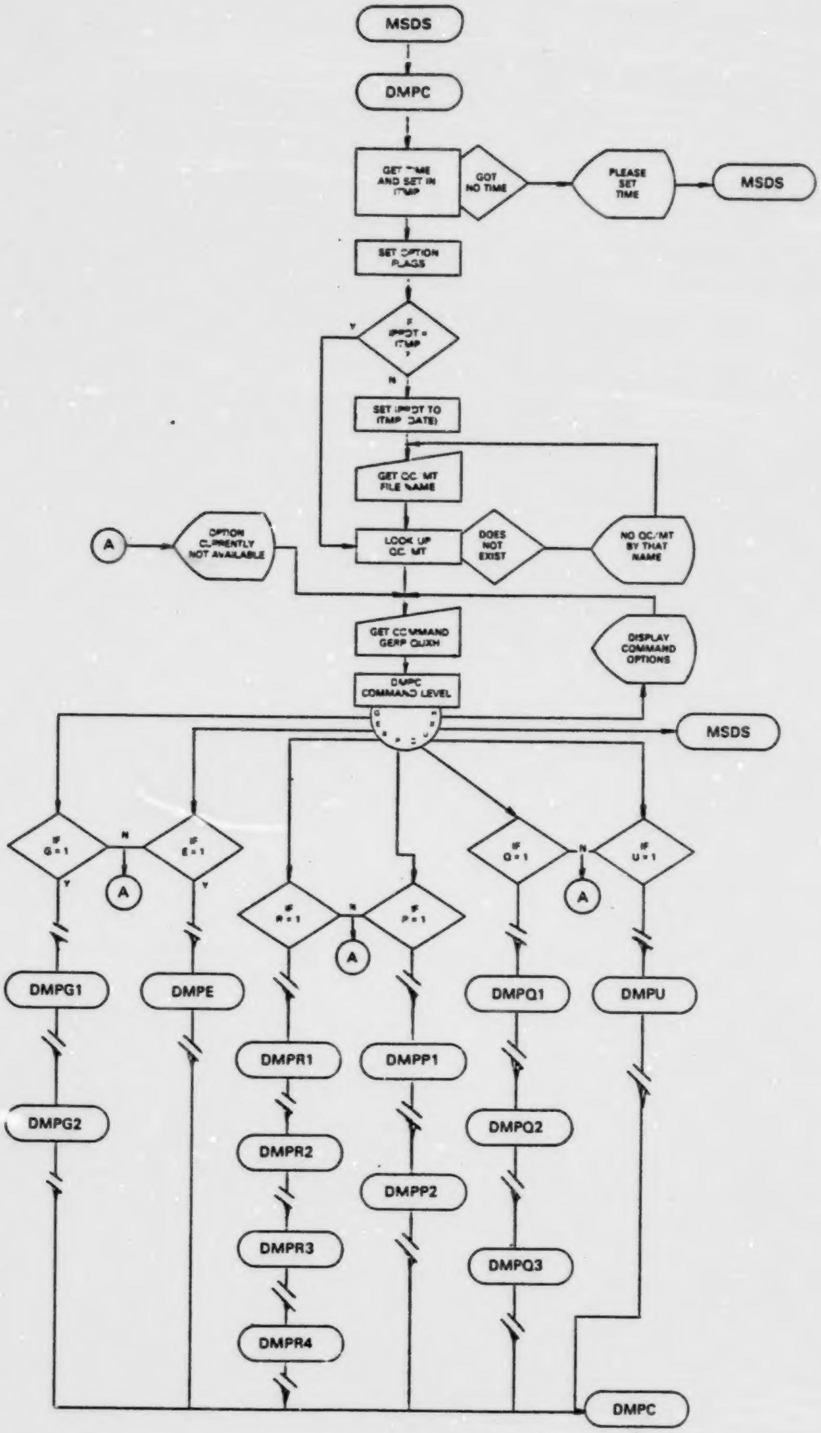

MSDS
DMPC
GET TIME AND SET IN ITMP
GOT NO TIME
PLEASE SET TIME
MSDS
SET OPTION FLAGS
IF IPPDT = ITMP ?
Y
N
SET IPPDT TO ITMP DATE)
GET QC MT FILE NAME
A
OPTION CURRENTLY NOT AVAILABLE
LOOK UP QC MT
DOES NOT EXIST
NO QC/MT BY THAT NAME
GET COMMAND GERP QUXH
DISPLAY COMMAND OPTIONS
DMPC COMMAND LEVEL
G E R P Q U X H
MSDS
IF G = 1
N
IF E = 1
Y
A
Y
IF R = 1
N
IF P = 1
A
IF Q = 1
N
IF U = 1
A
DMPG1
DMPE
DMPQ1
DMPU
DMPR1
DMPP1
DMPG2
DMPQ2
DMPR2
DMPP2
DMPR3
DMPQ3
DMPR4
DMPC

Figure 4.1 Flow Chart of the DMPC Program

TABLE 4.1

VARIABLES DESCRIPTIONS FOR THE DMPC PROGRAM

VARIABLE	DESCRIPTION
IANS(5)	answer string
ICH	channel number
ICMDST(4)	DMP command superstring
ICOMND	user command character
IDAY	day
IE	edit option flag
IEE	I/O error code
IEXTC(2)	QC/MT file extension
IG	report generator option flag
IHR	hour
ILEN	string length
IMIN	minute
IMO	month
IP	SFC results packaging option flag
IPRDT(4)	process date
IQ	QC/MT creation/edit option flag
IQCFN(10)	QC/MT file specifier
IQCNM(4)	QC/MT name (PARMETH)
IR	replicate statistics option flag
ISEC	seconds
ITMP(20)	text string
IU	utility option flag
IX	dummy integer
IYR	year
TTO	terminal I/O channel

SECTION 5

5. DMPQ - THE QUALITY CONTROL/METHOD TEMPLATE EDITOR PROGRAM

The DMPQ module is composed of three programs. The first (DMPQ1) creates new template skeletons with defaulted parameters. The second (DMPQ2) allows for QC/MT editing and the reporting of the file contents. The third program (DMPQ3) facilitates the update of quality control limits with data provided by the SFC system. (The QC/MT file is described in Section 11.4.)

5.1 QMPQ1 - THE SKELETON QC/MT CREATOR PROGRAM

The DMPQ1 program first gets the time and initializes variables. It then prompts the analyst for the QC/MT name. If there is no template with the given name, the program assumes that a new template is being created. A message is generated to the user to this effect. (If the user identifies a template which exists, DMPQ1 immediately chains to the DMPQ2 program which facilitates editing and report generation.)

To create a new template, the DMPQ1 program requires a quantitation file ($.QN, See Section 11.2) and a set of default parameters. The analyst is prompted for the quantitation file name. The program opens the file in the read mode and proceeds to count the compound names. The quantitation segment of the report is then read and examined for internal standard references. These references are stored in IARRAY. It is assumed that calibration standard quantitation report files will be used, however, the program does search for compounds identified as 'not found.' An attempt to correct the internal standard reference for an unfound compound is made by the program. Another check is made on the compound number to certify an uninterrupted sequence of compounds. Once these checks are made, control shifts to the default parameter entry segment.

The analyst is prompted for a set of default limit parameters which will be assigned to all of the compounds in the template. Hard-coded defaults are set and then a series of prompts are made to accept or change the defaults. After the defaults have been defined, the program prompts the analyst to accept or reject them. Once the defaults are accepted, the new QC/MT is opened in the write mode and the header is written. (The header contains parameters which allow for the absolute addressing of any record in the file.) The quantitation file is then opened in the read mode and the file is searched for the beginning of the compound name section. Each compound number and name is read and a record is written to the QC/MT file. The use code (IUSE) is set to zero unless IARRAY is the same as the compound number. This implies that the compound is an internal standard and a use code of 3 is assigned. Once the program has finished the creation task, it closes both the quantitation and the QC/MT files and chains to the DMPQ2 program for editing and reporting of the template contents.

Figure 5.1 shows the flow of the DMPQ1 program and Table 5.1 identifies the variables used in the program.

5.2 DMPQ2 - THE QC/MT EDITOR AND REPORT GENERATOR PROGRAM

The DMPQ2 program is a complex option oriented program which facilitates QC/MT editing and reporting. It also can access DMPQ1 in order to process

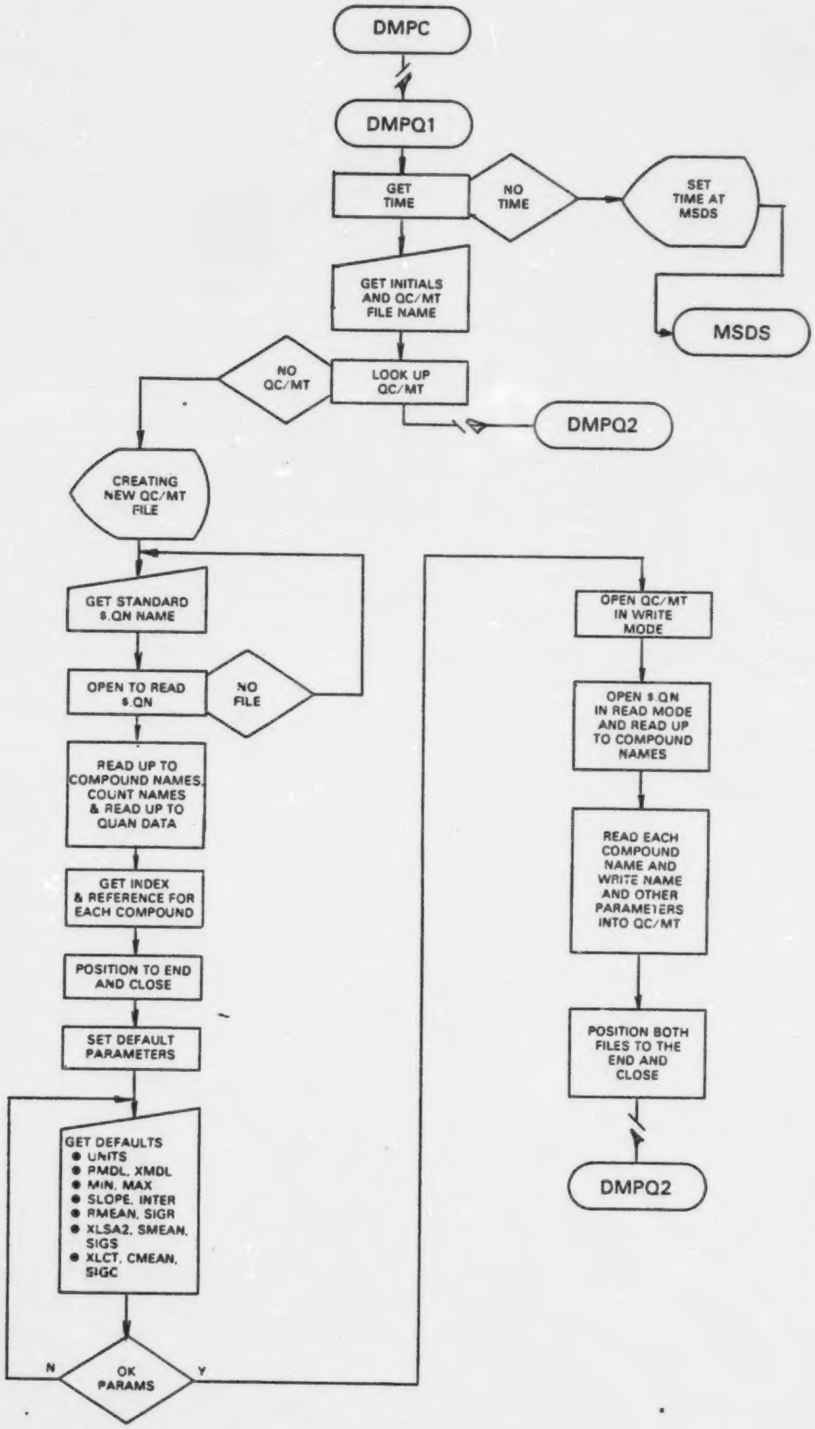

DMPC
DMPQ1
GET TIME
NO TIME
SET TIME AT MSDS
MSDS
GET INITIALS AND QC/MT FILE NAME
LOOK UP QC/MT
NO QC/MT
DMPQ2
CREATING NEW QC/MT FILE
GET STANDARD $.QN NAME
OPEN TO READ $.QN
NO FILE
READ UP TO COMPOUND NAMES, COUNT NAMES & READ UP TO QUAN DATA
GET INDEX & REFERENCE FOR EACH COMPOUND
POSITION TO END AND CLOSE
SET DEFAULT PARAMETERS
GET DEFAULTS
● UNITS
● RMDL, XMDL
● MIN, MAX
● SLOPE, INTER
● RMEAN, SIGR
● XLSA2, SMEAN, SIGS
● XLCT, CMEAN, SIGC
N
OK PARAMS
Y
OPEN QC/MT IN WRITE MODE
OPEN $.QN IN READ MODE AND READ UP TO COMPOUND NAMES
READ EACH COMPOUND NAME AND WRITE NAME AND OTHER PARAMETERS INTO QC/MT
POSITION BOTH FILES TO THE END AND CLOSE
DMPQ2

Figure 5.1 Flow Control of the DMPQ1 Program

TABLE 5.1

VARIABLES DESCRIPTIONS FOR THE DMPQ PROGRAMS

VARIABLE	DESCRIPTION
A	method range minimum
B	method range maximum
C	duplicate limit - slope
CMEAN	mean control standard recovery (%)
D	duplicate limit - intercept
D1 - D4	rounding constants
I	counter
I1	string position pointer
I2	string position pointer
IANS	answer string
IARRAY(*)	internal standard reference array
IBIT	a shiftable one bit mask
ICAS(6)	the compound CAS number string
ICDT(4)	QC/MT creation date
ICH	the QC/MT channel
ICHN	dummy I/O channel
ICINT(3)	QC/MT creators initials
IDAY	day
IDEF	default integer
IE	error code
IEE	I/O error variable
IEXTC(2)	QC/MT filename extension
IEXTN(2)	QUAN filename extension
IH1	hour
IHR	hour
ILEN	string length
ILOC	string location variable
IM1	minute
IMIN	minute
IMO	month
IND	the QUAN file compound counter
INIT(3)	initials of analyst
INM	a compound counter
INMS	number of compound in list
INSTP	edit range stop record
INSTR	edit range start record
IOPT	field edit option byte (1 bit per subfield)
IP1	string locator
IPRDT(4)	process date
IQCDSC(11)	QC/MT filename descriptor
IQCFN(8)	QC/MT filename specifier
IQCNM94)	QC/MT name (PARMETH)
IQCFN(11)	the QUAN file specifier
IQST3(3)	QC/MT edit command superstring
IREF	the internal standard reference number
IS1	second
ISBFLD	edit subfield counter

TABLE 5.1 (continued)

VARIABLE	DESCRIPTION
ISEC	second
ISEQ	the last internal standard reference number
ISTOP	selected stopping record for edit
ISTRT	selected start for edit
ITMP(40)	a text string
ITXT(70)	a text string
IUDT(4)	QC/MT date of last update
IUINT(3)	QC/MT update initials
IUPT(32)	use code description superstring
IUSE	compound use code
IX	dummy integer
IYR	year
KCHAN	dummy I/O channel
KEYC	source of control standard limits
KEYD	source of duplicate limits
KEYR	source of reagent blank limits
KEYS	source of spike limits
KUNITS(3)	the units of measure
LCH	the quantitation file channel
LOOK5(5)	QUAN report compound name header segment
LOOK6(4)	QUAN report quantification header segment
LOOK7(5)	'NOT FOUND' string
LPT	the lineprinter channel
MAXN	the maximum number of compounds in set
MOP	QC/MT edit option number
NAME(36)	the compound name string
NHOFF	QC/MT header offset (bytes)
NIND	the number of compounds
NM	number of compounds in QC/MT
NO	QC/MT header offset (bytes)
NR	number of bytes per QC/MT record
NREC	QC/MT record counter
NRECS	number of QC/MT records
NUMBS(5)	subfield selection superstring
RMDL	reagent method detection level
RMEAN	reagent blank mean
SIGC	control standard recovery standard deviation
SIGR	reagent blank standard deviation
SIGS	spike recovery standard deviation
SMEAN	mean spike recovery (%)
TTO	the terminal I/O channel
X1 - X10	rounded report variables
XLCT	known control standard amount
XLSA2	known spike amount
XMDL	matrix method detection level
XTMP	temporary or default real variable
Y1 - Y5	rounded report variables

another QC/MT file or it can access the DMPQ3 program in order to perform QC limit updating with SFC-supplied limits.

The DMPQ2 program first gets the date, opens the QC/MT file, and reads the header. The program then uses a subroutine call to display the header on the terminal. The analyst is prompted for an option and, on receiving a legitimate option, the program routes control to the appropriate program or DMPQ2 routine.

The QC/MT modification segment first prompts the analyst to identify the fields which are to be edited. Any of the eight primary groups may be selected. Each selection is recorded as a bit in variable IOPT. When no more fields are desired, the program prompts for the range of compounds affected by the edit. The default range of compounds contains the entire list.

Once the fields and records have been identified, the program proceeds to open the QC/MT file, position the pointer to the appropriate record, and read the QC/MT record to be affected. The pointer is then positioned to the end of file and the file is closed. It then uses a subroutine call to display the record contents. The program then prompts the user for the new information required for the selected fields. Checking is performed on CAS number check digits. In addition, no negative limit or amount variable is accepted. Once all of the selected subfields have been dealt with, the parameters are redisplayed and the analyst can accept or reject them. Once the data are acceptable, it is written into the file. This is done by opening the field in the modify mode, positioning the pointer to the appropriate record, writing the record, positioning the pointer to the end of file and closing the file.

After all the selected records have been edited, the program modifies the QC/MT header record to indicate that the file was modified. The program then returns to the option prompt.

If the report option (R) is selected, the program opens the QC/MT file and positions it to the first record. It then deletes and opens the QC/MT report file (DMPRPTQ.99) in the write mode and writes the report header. It then reads each record and writes it to the report file. The same write routines and formatting used during terminal displays are used in the report writing. Records are counted and the report is paged at appropriate intervals. This is done because of the semipermanent nature of the QC/MT. When all of the records have been reported, the QC/MT file is closed and the report file is copied to the line printer. The program then returns to the option prompt.

If the analyst elects to get a new QC/MT file name (G), the program chains to DMPQ1. If the analyst wishes to exit (X), the program chains to the DMPC program. Finally, if the SFC limit update option (S) is selected, the DMPQ2 program chains the DMPQ3 program.

Figure 5.2 shows the flow of the DMPQZ program and Table 5.1 identifies the variables used in the program.

5.3 DMPQ3 - THE QC/MT LIMIT UPDATE PROGRAM

The DMPQ3 program will extract limits from an SFC limit record file and write them into the selected QC/MT file. The analyst will be responsible for transferring the limit file from the SFC system to the INCOS system via magnetic tape. (Section 11.9 describes the form and content of such a file.) The DMPQ3 program prompts the user for the full SFC limit file name. The program opens the file in the read mode and examines the file for

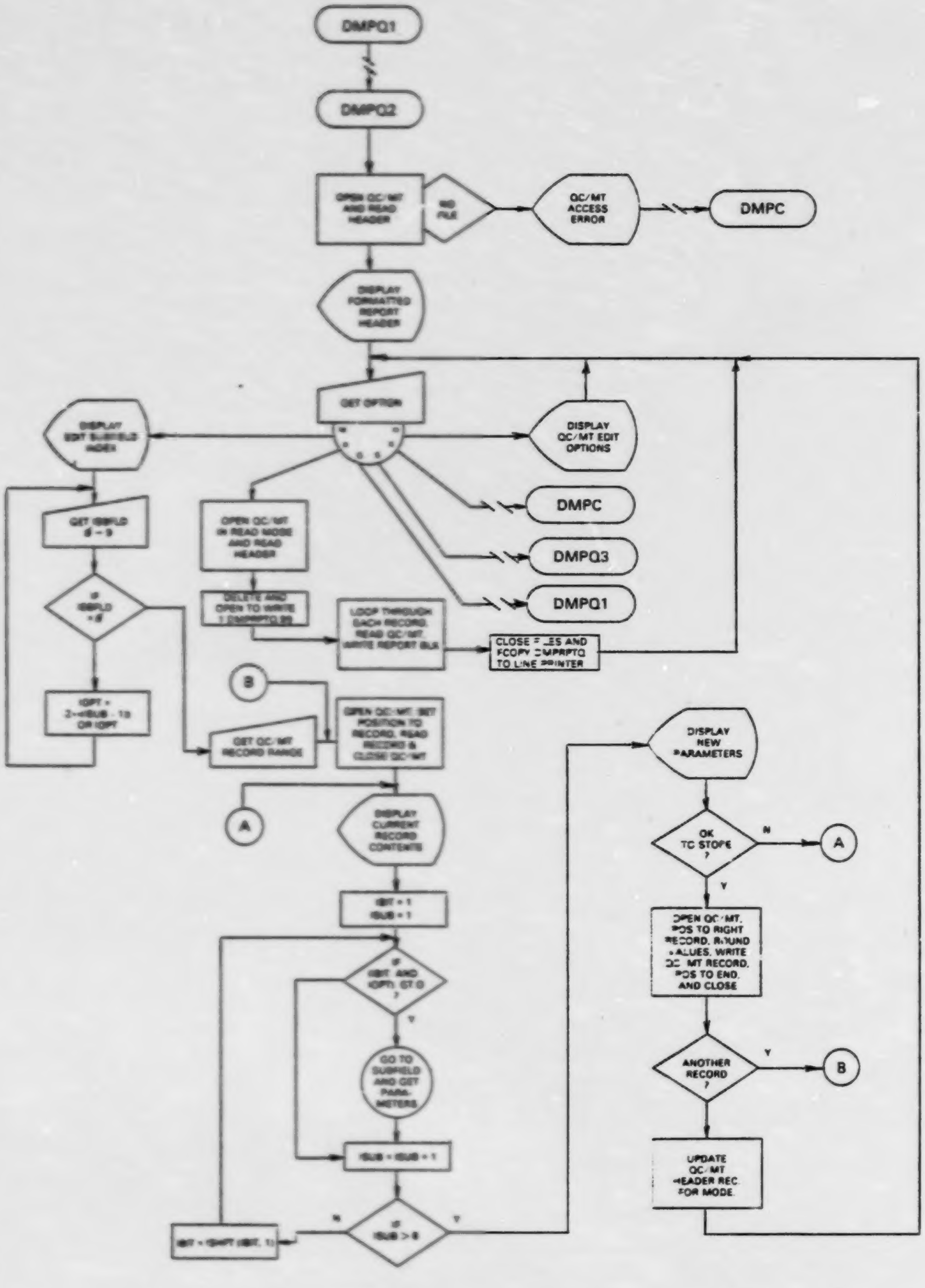

DMPQ1
DMPQ2
OPEN QC/MT AND READ HEADER
NO FILE
QC/MT ACCESS ERROR
DMPC
DISPLAY FORMATTED REPORT HEADER
GET OPTION
DISPLAY EDIT SUBFIELD INDEX
DISPLAY QC/MT EDIT OPTIONS
DMPC
DMPQ3
DMPQ1
GET ISBFLD Ø – 9
IF ISBFLD = Ø
IOPT = 2**(ISUB – 1) OR IOPT
OPEN QC/MT IN READ MODE AND READ HEADER
DELETE AND OPEN TO WRITE 1 DMPRPTQ.99
LOOP THROUGH EACH RECORD, READ QC/MT, WRITE REPORT BLK
CLOSE FILES AND FCOPY DMPRPTQ TO LINE PRINTER
B
GET QC/MT RECORD RANGE
OPEN QC/MT, SET POSITION TO RECORD, READ RECORD & CLOSE QC/MT
A
DISPLAY CURRENT RECORD CONTENTS
IBIT = 1
ISUB = 1
IF IBIT AND IOPT) GT 0 ?
Y
GO TO SUBFIELD AND GET PARAMETERS
ISUB = ISUB + 1
N
IF ISUB > 8
Y
IBIT = ISHFT (IBIT, 1)
DISPLAY NEW PARAMETERS
OK TO STORE ?
N
A
Y
OPEN QC/MT, POS TO RIGHT RECORD, ROUND VALUES, WRITE QC/MT RECORD, POS TO END, AND CLOSE
ANOTHER RECORD ?
Y
B
UPDATE QC/MT HEADER REC. FOR MODE

Figure 5.2 Flow Chart of the DMPQ2 Program

a specified header sequence. If the file is not found or end-of-file is detected before the sequence is found, the program prompts for another file name. Once an appropriate file has been identified, the QC/MT file is opened in the modify mode. The name and CAS number are read out of the QC/MT file and the SFC limit file is then searched for the occurrence of that CAS number. Once a match has been found, the limit type is identified and the appropriate number of records are read. The program will only update the QC/MT file if the SFC supplied limits were computed on the basis of real data. Once the QC/MT fields have been modified, the SFC limit file is read again to determine if more limits are contained for the same CAS numbered compound. If so, the limit type is identified and the same QC/MT record is modified for a different limit set. All SFC supplied limits will be grouped by CAS number but may be in a different order from the QC/MT file. Once a QC/MT limit record has been updated, the SFC limit file is positioned to the first record, and the next QC/MT name and CAS number are read. The search is made again of the SFC limit file in order to match the CAS numbers. The QC/MT record is updated and the process continues until all QC/MT records have been serviced. This task obviously takes a long time to complete. Once the process is complete, the DMPQ3 program chains the DMPQ2 program for potential editing and reporting.

The DPMP3 program has not been implemented.

SECTION 6
DMPG - THE DATA CONSOLIDATOR AND REPORT GENERATOR PROGRAMS

The DMPG module consists of two large programs: DMPG1 and DMPG2. DMPG1 gathers information from the sample log file (Section 11.1), the sample quantitation file (Section 11.2), and the sample qualitative identification file (Section 11.3), and forms a reduced data file (Section 11.5). (Some variations of the reduced data file occur based on the sample type.) The DMPG2 program uses the reduced data file and a QC/MT file (Section 11.4) to generate specialized measurement and quality control reports on the sample results. Both programs can be namelist driven and DMPG2 can be executed directly from MSDS.

6.1 DMPG1 - THE REDUCED DATA FILE GENERATOR

The DMPG1 program initializes program variables and gets the date from the system clock. It then prompts the user for the QC/MT name. The program then prompts for the file name input mode. If it is 'L', then the namelist file name is requested. If the file name is to be entered manually ('M'), the program prompts for the sample name. If a name list is being used, the DMPG1 program will process all entries before transferring control to the report generator (DMPG2).

For each sample name, the DMPG1 program determines the sample type (ICODE). If the terminating character of the sample name is not U,D,F,S,R,N, or D, the program assigns a code of 6 indicating a non-SFC (N) sample. The program then certifies that a log file is present for the sample and reads the log file header information. The program searches for the '$$$' string and then reads the volume and data basing parameters. If the '$$$' string is not present, an error message is generated and processing continues on to the next namelist sample name.

Once the log file has been read, the quantitation file is opened in the read mode. The compound name segment is found by searching for the 'NO NAME' subheader string. The GC/MT file is then opened and each name in the quantitation report is compared to each name in the QC/MT file. Next the quantitation data is read and put into the SDATA and IDATA arrays. The reduced data file is then opened in the write mode and the header record is written. The program then computes and writes the data on each compound into the file. The relative retention time is put into integer form for storage in the reduced data file.

If the sample type is an unknown (U), then the program executes a special process to determine whether a duplcate and/or spiked sample measurement was made on aliquots of the unknown. The 'U' in the sample identifier is changed to a 'D' and the program checks to see if the log file and the quantitation files exist for the sample name. If so, those files are opened and read in a similar manner to the way the original sample files were processed. The duplicate data is appended to the reduced data file and the main header is updated to indicate that the file contains duplicate data. (ICODE is changed from 0 to 1.) The same sequence is conducted for the spiked unknown whose code is 'F'. The resulting reduced data file can contain one of four codes: ICODE = 0 indicates an unattached unknown; ICODE = 1 indicates an unknown and its duplicate; ICODE = 2 indicates an unknown and its spike; and ICODE = 3 indicates an unknown which has been both duplicated and run as a spiked sample.

The DMPG1 program is designed to search for qualitative identification files for unknown type samples (U) and for non-SFC type samples (N). If a qualitative identification file is found for either of these sample types, it is appended to the reduced data file.

Once the reduced data file has been created and filled, it is closed and the DMPG1 program chains the DMPG2 program.

Figure 6.1 shows the flow of the DMPG1 program and Table 6.1 identifies the variables used in the program.

6.2 DMPG2 - THE MEASUREMENT AND QUALITY CONTROL ASSESSMENT REPORTING PROGRAM

The DMPG2 program first initializes variables and gets the date from the system clock. It then checks the sample name input mode (M or L). If the mode is not recognized as M or L, the program assumes that entry was from some place other than the DMPG1 program. It then prompts the analyst for the name input mode and the subsequent file name or namelist.

Once a sample file has been identified, the program opens the reduced data file and reads the header record. The QC/MT file name is reconstructed and checked for existence. The template is then opened in the read mode and the header is read. The report file (Section 11.6) is deleted and then opened in the write mode and the report header is generated for the primary sample. Depending on the sample type code (ICODE), the program branches to one of four report subheader segments and generates the subheader. It then reads corresponding records from both the QC/MT file and the reduced data file and, depending on the sample code, generates one or more report lines. As unknown (U) analytes are reported, the SDATA array is used to store

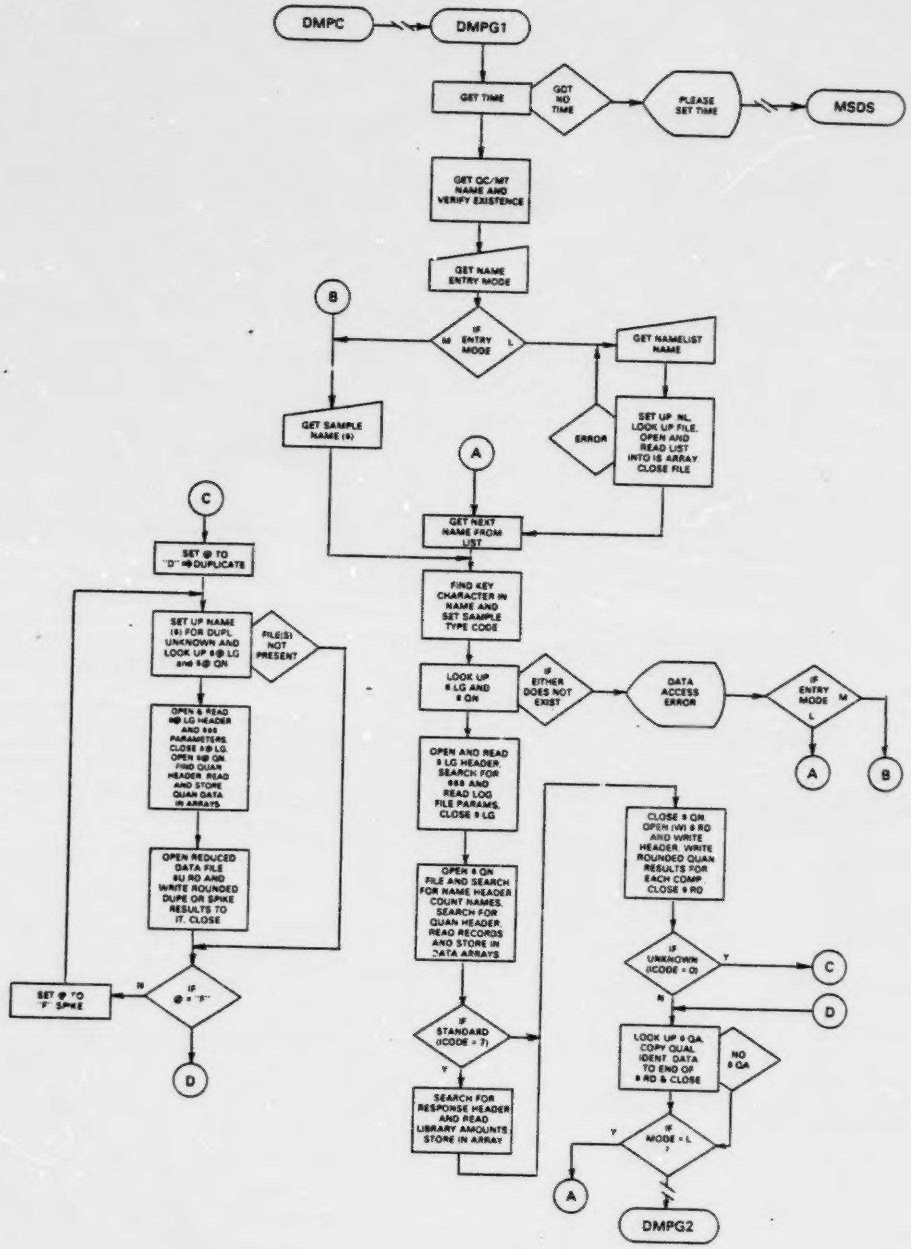

DMPC
DMPG1
GET TIME
GOT NO TIME
PLEASE SET TIME
MSDS
GET QC/MT NAME AND VERIFY EXISTENCE
GET NAME ENTRY MODE
B
IF ENTRY MODE
M
L
GET NAMELIST NAME
ERROR
SET UP NL. LOOK UP FILE. OPEN AND READ LIST INTO IS ARRAY. CLOSE FILE
GET SAMPLE NAME (#)
A
GET NEXT NAME FROM LIST
FIND KEY CHARACTER IN NAME AND SET SAMPLE TYPE CODE
LOOK UP # LG AND # QN
IF EITHER DOES NOT EXIST
DATA ACCESS ERROR
IF ENTRY MODE
M
L
A
B
OPEN AND READ # LG HEADER. SEARCH FOR ### AND READ LOG FILE PARAMS. CLOSE # LG
OPEN # QN FILE AND SEARCH FOR NAME HEADER COUNT NAMES. SEARCH FOR QUAN HEADER. READ RECORDS AND STORE IN DATA ARRAYS
IF STANDARD (ICODE = 7)
Y
SEARCH FOR RESPONSE HEADER AND READ LIBRARY AMOUNTS STORE IN ARRAY
CLOSE # QN. OPEN (W) # RD AND WRITE HEADER. WRITE ROUNDED QUAN RESULTS FOR EACH COMP. CLOSE # RD
IF UNKNOWN (ICODE = 0)
Y
C
N
D
LOOK UP # QA. COPY QUAL IDENT DATA TO END OF # RD & CLOSE
NO # QA
IF MODE = L ?
Y
A
DMPG2
C
SET @ TO "D" ⇒ DUPLICATE
SET UP NAME (#) FOR DUPL UNKNOWN AND LOOK UP #@ LG and #@ QN
FILE(S) NOT PRESENT
OPEN & READ #@ LG HEADER AND ### PARAMETERS. CLOSE #@ LG. OPEN #@ QN. FIND QUAN HEADER. READ AND STORE QUAN DATA IN ARRAYS
OPEN REDUCED DATA FILE #U RD AND WRITE ROUNDED DUPE OR SPIKE RESULTS TO IT. CLOSE
IF @ = "F"
N
SET @ TO "F" SPIKE
D

Figure 6.1 Flow Chart of the DMPG1 Program

TABLE 6.1

VARIABLES DESCRIPTIONS FOR THE DMPG PROGRAMS

VARIABLE	DESCRIPTION
A	method range minimum
AM	computed amount found on QUAN report
AR	area
B	method range maximum
C	duplicate limit - slope
CMEAN	mean control standard recovery (%)
CONC	concentration
CONCO	original concentration for spikes and duplicates
D	duplicate limit - intercept
D1 - D4	rounding constants
DIFF	duplicate difference
F	same as factor
FACTOR	concentration multiplier
I	counter
I1	string position pointer
I2	string position pointer
I3	a counter
IACCNT(4)	account code or SFC sample identifier
IACDT(4)	acquisition date for primary sample
IACDT2(4)	acquisition date for spike or duplicate
IANLST(4)	analyst name or initials
IANS	answer string
IC	number of names in namelist
ICAS(6)	compound CAS number string
ICD	duplicate or spike code
ICH	file input channel number
ICHK	ICODE check digit
ICKST(5)	name of reference control standard
ICMNT(32)	DMP user comment
ICODE	sample type code
ICOMP(64)	computation equation from QUAN report
ICOND(32)	comment on measurement conditions
ICP	sample code counter for spikes and duplicates
ICRDF	'D' or 'F' substitution character
ICREQ	'='
ICRQS	'µ'
ICRU	'U'
IDATA(*,1)	internal standard reference numbers
IDATA(*,2)	integerized relative retention times
IDAY	day
IDT	internal standard reference
IE	error code
IEE	I/O error variable
IER	error code
IEXDT(4)	extraction date for primary sample
IEXDT2(4)	extraction date for duplicate or spike
IEXTA(2)	qualitative identification file extension

TABLE 6.1 (Continued)

VARIABLE	DESCRIPTION
IEXTC(2)	QC/MT filename extension
IEXTG(2)	log file extension
IEXTN(2)	QUAN filename extension
IEXTR(2)	reduced data file extension
IFLDSC(11)	general DMP file descriptor
IFLFN(10)	general DMP file specifier
IFLNM(8)	general DMP file name
IHR	hour
ILEN	string length
IMIN	minute
IMO	month
INO	the QUAN file compound counter
INLDSC(11)	namelist file descriptor
INLNM(8)	namelist file name
INM	a compound counter
INST(3)	instrument name
IP1	string locator
IP2	a string position variable
IP3	a string position variable
IPMCT	reduced data file compound count
IPMTH(4)	fixed list PARMETH code
IPOS	a string position variable
IPR	a string position variable
IPRDT(4)	process date
IPVTH(4)	variable list PARMETH code
IQCDSC(11)	QC/MT filename descriptor
IQCFN(8)	QC/MT filename specifier
IQCMS	QC message string
IQCNM(4)	QC/MT name (PARMETH)
IREF	the internal standard reference number
IRGBK(5)	name of reference reagent blank
IRPDT(4)	report date
IRRDT(4)	run result formation date
IRRNM(9)	SFC run result file name
IRUN(10)	run identifier
IS(11,MAXSM)	namelist name array
ISAMP	namelist sample counter
ISEC	second
ISMPLE(32)	comment on sample
ISOURCE	sample media code
ITMP(20)	a text string
ITXT(5)	a text string
ITYPE	duplicate/spike character code
IUSE	compound use code
IVS(2)	sample media indicator
IX	dummy integer
IYR	year
KEYC	source of control standard limits
KEYD	source of duplicate limits

TABLE 6.1 (Continued)

VARIABLE	DESCRIPTION
KEYR	source of reagent blank limits
KEYS	source of spike limits
KO	QUAN file error flag
KP1	string position pointer
KUNITS(3)	the units of measure
KVOL(2)	sample volume indicator
LCH	file I/O channel number
LIMIT	BDL/GTL notification flag
LN	length of sample name (location of type code)
LNN	LN+1
LOC	string location variable
LOOK5(5)	QUAN report compound name header segment
LOOK1(4)	'ACQUIRE' in log file header
LOOK2(2)	"$$$' before log file parameters
LOOK3(5)	'NOT FOUND' in QUAN file records
LOOK4(3)	'AMOUNT' in QUAN file
LOOK6(4)	QUAN report quantification header segment
LP	a string length variable
LP1	a string length variable
LPT	the lineprinter channel
LSTMD	list mode
LTX	text string length
LTYP	sample type variable
MATCH(20)	sample media super string
MAXN	maximum number of compounds in set
MAXSM	maximum number of names in namelist
MCH	report channel
MCODE(4)	sample type super string
METH(2)	peak measurement mode
MUNITS(20)	the units super string
NAME(36)	the compound name string
NCOMP(2)	computation mode indicator
NIND	the number of compounds
NM	number of compounds in QC/MT
NN	number of bytes per QC/MT record
NO	QC/MT header offset (bytes)
NPIND	duplicate or spike compound index
NSAMP	number of samples in namelist
PRR	percent relative range
RC	critical range difference for duplicates
RMDL	reagent method detection level
RMEAN	reagent blank mean
RRT	relative retention time
SDATA(*,1)	measured area
SDATA(*,2)	computed concentration
SIGC	control standard recovery standard deviation
SIGR	reagent blank standard deviation
SIGS	spike recovery standard deviation

TABLE 6.1 (Continued)

VARIABLE	DESCRIPTION
SMEAN	mean spike recovery (%)
TTO	the terminal I/O channel
VEX	extract volume
VINJ	injection volume
VOL	sample volume
VX	dummy volume
X	dummy real number
X1 - X8	rounded report variables
XLCT	known control standard amount
XLSA2	known spike amount
XMDL	matrix method detection limit

concentration values for use in subsequent duplicate and spike recovery reports.

When all of the analytes have been reported for a standard (S), a control standard (C) or a reagent blank (R), the report is output to the lineprinter and the next sample in the namelist is processed. If the sample is an unknown and has additional duplicate and/or spike data, then the subheader record is read from the reduced data file and the QC/MT pointer is positioned just after the template header. A new report header and subheader are generated to the report file for the spike or duplicate. Concurrent readings are then made from both the template and reduced data files and the appropriate report line is generated to the report file. The variable ICHK is used to determine whether to examine the reduced data file for duplicate and/or spike data.

During the processing of unknowns (U), spiked unknowns (F), duplicate unknowns (D), and non-SFC samples (N), the program checks for the presence of a surrogate spike compound. This is identified by a QC/MT use code of IUSE = 2. If a surrogate is detected, a mini report is generated on surrogate spike recovery and a quality control assessment is made on the basis of stored limits.

Once the quantitative data has been processed, control is routed to the qualitative identification segment of the program. Only primary unknowns (U) and non-SFC samples (N) are examined for qualitative data. If qualitative data is detected at the end of the reduced data file, a subheader is generated to the report file and the information is copied in text form to the report file. After the report file has been generated, it is output to the lineprinter and the program continues to process other files in the namelist. When the DMPG2 program is done with all of the samples in the list, it chains to DMPC.

Figure 6.2 shows the flow of the DMPG2 program and Table 6.1 identifies the variables used in the program.

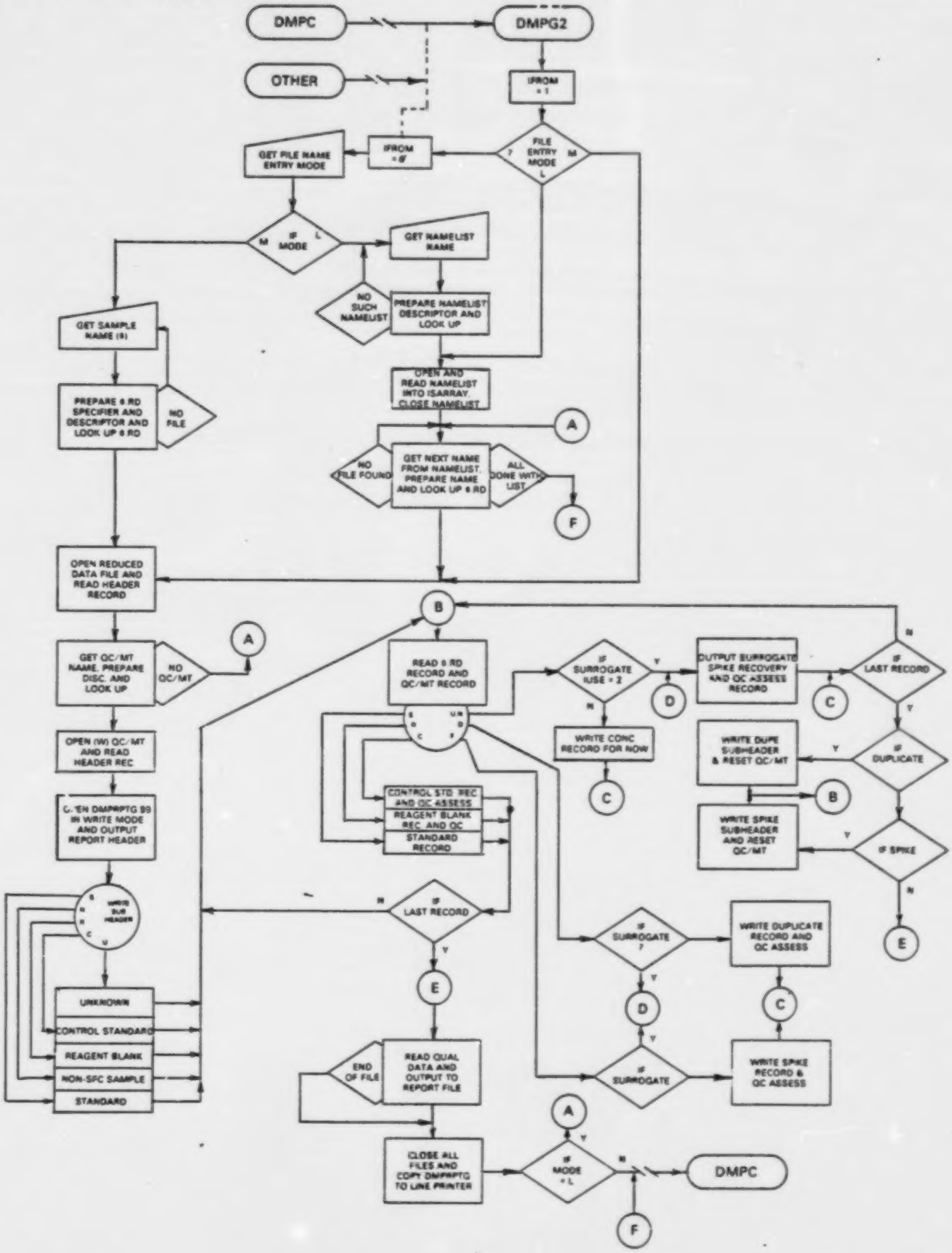

DMPC
DMPG2
OTHER
IFROM = 1
FILE ENTRY MODE
?
M
L
IFROM = 0
GET FILE NAME ENTRY MODE
IF MODE
M
L
GET NAMELIST NAME
NO SUCH NAMELIST
PREPARE NAMELIST DESCRIPTOR AND LOOK UP
GET SAMPLE NAME (S)
PREPARE 6 RD SPECIFIER AND DESCRIPTOR AND LOOK UP 6 RD
NO FILE
OPEN AND READ NAMELIST INTO ISARRAY, CLOSE NAMELIST
A
NO FILE FOUND
GET NEXT NAME FROM NAMELIST, PREPARE NAME AND LOOK UP 6 RD
ALL DONE WITH LIST
F
OPEN REDUCED DATA FILE AND READ HEADER RECORD
B
GET QC/MT NAME, PREPARE DISC. AND LOOK UP
NO QC/MT
A
READ 6 RD RECORD AND QC/MT RECORD
S
D
C
U,N
D
F
IF SURROGATE IUSE = 2
Y
N
D
OUTPUT SURROGATE SPIKE RECOVERY AND QC ASSESS RECORD
C
IF LAST RECORD
N
Y
WRITE CONC RECORD FOR NOW
C
WRITE DUPE SUBHEADER & RESET QC/MT
IF DUPLICATE
Y
B
WRITE SPIKE SUBHEADER AND RESET QC/MT
IF SPIKE
Y
N
E
OPEN (W) QC/MT AND READ HEADER REC
OPEN DMPRPTG 99 IN WRITE MODE AND OUTPUT REPORT HEADER
CONTROL STD. REC AND QC ASSESS
REAGENT BLANK REC. AND QC
STANDARD RECORD
WRITE SUB HEADER
S
R
N
C
U
IF LAST RECORD
N
Y
IF SURROGATE ?
WRITE DUPLICATE RECORD AND QC ASSESS
UNKNOWN
CONTROL STANDARD
REAGENT BLANK
NON-SFC SAMPLE
STANDARD
E
D
Y
C
Y
IF SURROGATE
WRITE SPIKE RECORD & QC ASSESS
END OF FILE
READ QUAL DATA AND OUTPUT TO REPORT FILE
A
Y
CLOSE ALL FILES AND COPY DMPRPTG TO LINE PRINTER
IF MODE = L
N
DMPC
F

Figure 6.2 Flow Chart of the DMPG2 Program

SECTION 7
DMPE - THE REDUCED DATA FILE EDITOR PROGRAM

The DMPE Program facilitates the modification of any reduced data file (Section 11.5). After the program obtains the date and initializes variables, it prompts for a file name. When it has certified that the reduced data file exists, it prompts for the record and field numbers to be modified. Any number of fields may be selected and each is represented by a bit in the IOPT variable. The record range is defaulted to include the entire list of compounds. If the file contains data from a duplicate and/or a spike, additional edit fields will be displayed showing the selectable fields. Because of the configuration of the reduced data file for unknown samples, editing proceeds first with the primary sample then with the duplicate, and finally with the spike data.

Once editing has been performed on the quantitative sample data, the program will scan the reduced data file for qualitative identification data. (This is done only for unknowns and non-SFC samples.) If qualitative data is detected in the file, the program prompts the analyst to identify the appropriate edit fields. The program then reads each record and prompts for user modifications. This process is not record selectable and will proceed until the end of file is reached. On completion of the edit, the program will optionally chain to the DMPG2 or the DMPC program.

The DMPE program has not been implemented.

SECTION 8
DMPR - THE REPLICATE STATISTICS COMPUTATION AND REPORT GENERATION PROGRAMS

The DMPR module consists of four medium sized programs which perform statistical processing. The DMPR1 program establishes the parameter of interest and the reduced data files (Section 11.5) to be examined. The DMPR2 program performs the statistical computations. The DMPR3 program generates the appropriate report. And the DMPR4 program uses replicate data to update specific parameters in the QC/MT file (Section 11.4).

8.1 DMPR1 - THE MODE SELECTION AND FILE IDENTIFICATION PROGRAM

After initializing parameters and getting the date, the DMPR1 program obtains the QC/MT file name from the user. The existence of the file is then certified. The analyst is prompted to select one of three modes by which replicate samples are to be identified. The keyboard option (K) requires manual entry of each name, while the list option (L) allows for namelist entry. If the namelist option is used, the program will get the name of the namelist, certify its existence, read its contents, and route control to the keyboard entry segment to allow for additional manually entered file names. A third option exists which allows for the "wild-card" (W) selection of sample names. The analyst is prompted to enter a root word containing a pound sign (#) indicating the position of a counting digit. The program finds the pound sign and generates ten file names with digits 0 through 9 in the pound sign's position. Additional file names may then be added through the keyboard entry segment as with the namelist option mentioned above.

The namelist entry session is terminated with a null response for another file name. The list of files is then examined to determine if each reduced data file exists and is of the right type. This check is made by opening each reduced data file and reading in the file header. The QC/MT file name in the reduced data file is checked against the analyst supplied name, and ICODE is checked against the sample file type (ITYPE) established with the first acceptable sample name. All file names are displayed as being either accepted or rejected and appropriate error messages are generated if the file is rejected. Rejected file names in the namelist are assigned 'NO' values in the eleventh array position of their descriptor.

At least three acceptable files are required before statistics can be calculated. If this condition is not met, the program is restarted. If enough files are present, the DMPR1 program chains to the DMPR2 program.

Figure 8.1 shows the flow of the DMPR1 program and Table 8.1 identifies the variables used in the program.

8.2 DMPR2 - THE REPLICATE STATISTICS COMPUTATION PROGRAM

The DMPR2 program first clears the AMAT array which is used for all statistical computations. The AMAT array collects numerical averages in column #1, standard deviation data in column #2, and a replicate count in column #3.

The main portion of the DMPR2 program is a two-pass loop which first accumulates mean values and then computes the sum of the squared differences from the mean. Each acceptable reduced data file is opened and the header is read. The compound records are then read and the appropriate value is

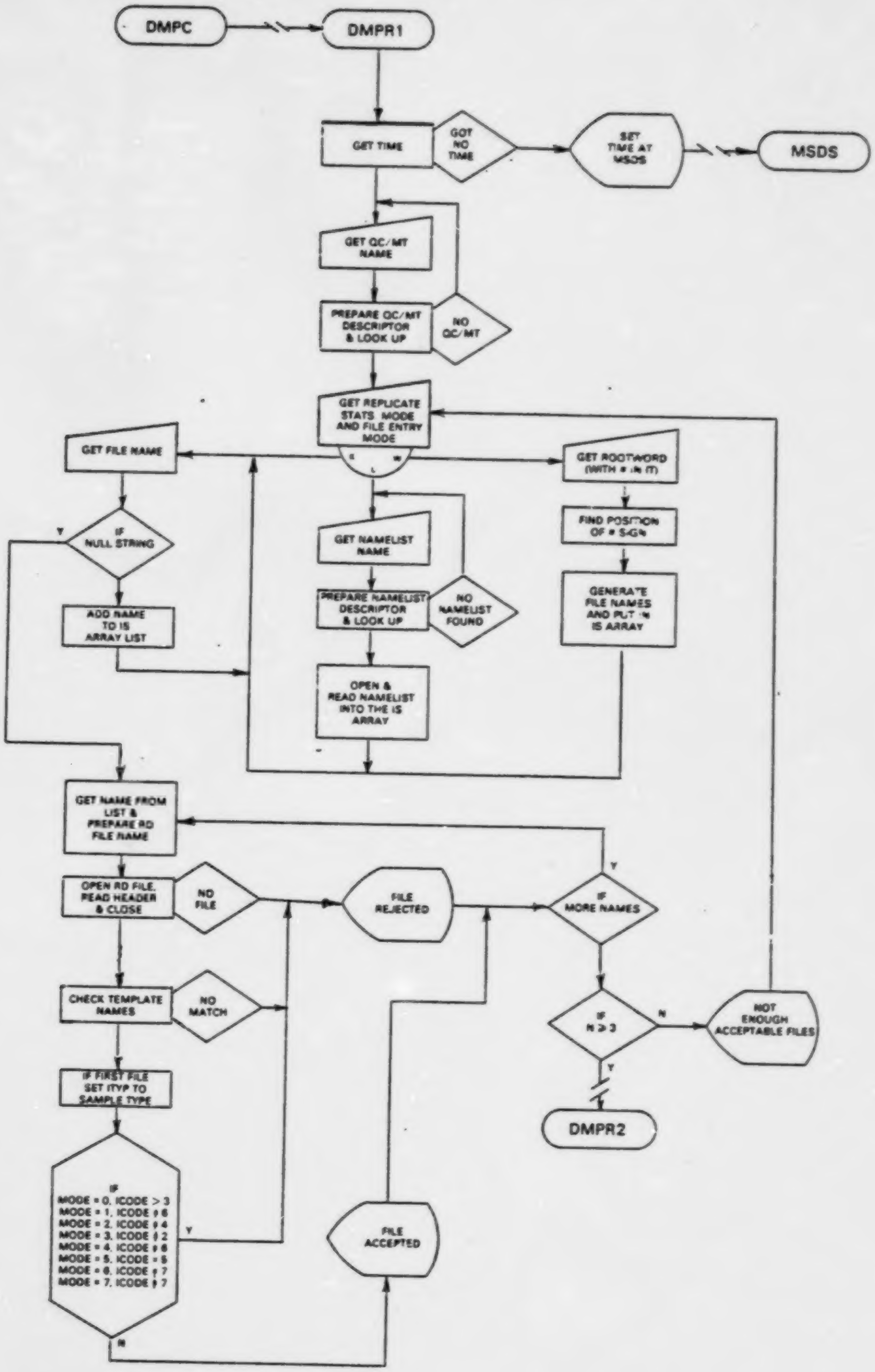

DMPC
DMPR1
GET TIME
GOT NO TIME
SET TIME AT MSDS
MSDS
GET QC/MT NAME
PREPARE QC/MT DESCRIPTOR & LOOK UP
NO QC/MT
GET REPLICATE STATS MODE AND FILE ENTRY MODE
GET FILE NAME
GET ROOTWORD (WITH # IN IT)
IF NULL STRING
Y
ADD NAME TO IS ARRAY LIST
GET NAMELIST NAME
PREPARE NAMELIST DESCRIPTOR & LOOK UP
NO NAMELIST FOUND
OPEN & READ NAMELIST INTO THE IS ARRAY
FIND POSITION OF # SIGN
GENERATE FILE NAMES AND PUT IN IS ARRAY
GET NAME FROM LIST & PREPARE RD FILE NAME
OPEN RD FILE, READ HEADER & CLOSE
NO FILE
FILE REJECTED
IF MORE NAMES
Y
CHECK TEMPLATE NAMES
NO MATCH
IF N ≥ 3
N
NOT ENOUGH ACCEPTABLE FILES
Y
DMPR2
IF FIRST FILE SET ITYP TO SAMPLE TYPE
IF
MODE = 0, ICODE > 3
MODE = 1, ICODE ≠ 6
MODE = 2, ICODE ≠ 4
MODE = 3, ICODE ≠ 2
MODE = 4, ICODE ≠ 6
MODE = 5, ICODE = 5
MODE = 6, ICODE ≠ 7
MODE = 7, ICODE ≠ 7
Y
N
FILE ACCEPTED

Figure 8.1 Flow Chart of the DMPR1 Program

TABLE 8.1

VARIABLES DESCRIPTIONS FOR THE DMPR PROGRAMS

VARIABLE	DESCRIPTION
A	method range minimum
AL	rounded mean
A2	rounded standard deviation
A3	rounded known amount
AMAT(*,1)	mean summing array
AMAT(*,2)	standard deviation summing array
AMAT(*,3)	replicate summing array
AR	area
B	method range maximum
C	duplicate limit - slope
CMEAN	mean control standard recovery (%)
CN	concentration
D	duplicate limit - intercept
DSK	channel number
F	same as FACTOR
FMDL	MDL assessment multiplier
I	counter
I1	string position pointer
I2	string position pointer
IACCNT(4)	account code or SFC sample identifier
IACDT(4)	acquisition date for primary sample
IACDT2(4)	acquisition date for spike or duplicate
IANLST(4)	analyst name or initials
IANS(5)	answer string
IC	number of names in namelist
ICAS(6)	compound CAS number string
ICD	duplicate or spike code
ICH	file input channel number
ICHAR	character variable
ICHR	a character variable
ICKST(5)	name of reference control standard
ICMNT(32)	DMP user comment
ICODE	sample type code
ICOND(32)	comment on measurement conditions
IDAY	day
IDT	internal standard reference
IEE	I/O error variable
IER	error code
IEXDT(4)	extraction date for primary sample
IEXDT2(4)	extraction date for duplicate or spike
IEXTC(2)	QC/MT filename extension
IEXTR(2)	reduced data file extension
IFLDSC(11)	general DMP file descriptor
IFLFN(10)	general DMP file specifier
IFLNM(8)	general DMP file name
IH1	hour
ILEN	string length

TABLE 8.1 (Continued)

VARIABLE	DESCRIPTION
IM1	minute
IMO	month
IND	the QUAN file compound counter
INDQ	QC/MT index counter
INLDSC(11)	namelist file descriptor
INLNM(8)	namelist file name
INST(3)	instrument name
IP	a compound counter
IPMTH(4)	fixed list PARMETH code
IPOS	a string position variable
IPRDT(4)	process date
IQCDSC(11)	QC/MT filename descriptor
IQCFN(8)	QC/MT filename specifier
IQCNM(4)	QC/MT name (PARMETH)
IQST1(5)	replicate mode selection super string
IQST2(3)	filename entry option super string
IRGBK(5)	name of reference reagent blank
IRRT	integerized relative retention time
IS(11,MAXSM)	namelist name array
IS1	second
ISMPLE(32)	comment on sample
ITEMS(64)	item of interest super string
ITMP(20)	a text string
ITP	a string pointer
ITXT950)	a text string
IUSE	compound use code
IX	dummy integer
IYR	year
KEYC	source of control standard limits
KEYD	source of duplicate limits
KEYR	source of reagent blank limits
KEYS	source of spike limits
KH*(#)	report subheader strings
KP1	string position variable
KUNITS(3)	the units of measure
LCH	file I/O channel number
LMD	the replicate name entry mode
LN	length of sample name (location of type code)
LNN	LN+1
LOOP	the mean/standard deviation loop counter
LPT	the lineprinter channel
M	number of compounds
M1	a string segment delimeter
M2	a string segment delimeter
MAXN	maximum number of compounds in set
MAXSM	maximum number of names in namelist

TABLE 8.1 (Continued)

VARIABLE	DESCRIPTION
MCH	report channel
MI	the replicate counter
MODE	replicate computation mode
MODEP	MODE+1
N	number of acceptable replicates
NO	number of acceptable replicates
NAME(36)	the compound name string
NCOMP(2)	computation mode indicator
NI	replicate counter
NIND	the number of compounds
NM	number of compounds in QC/MT
NN	number of bytes per QC/MT record
NO	QC/MT header offset (bytes)
NQ	counter
NX	acceptable replicate counter
RESPF(MAXN)	temporary array for response ratios and spike amounts
RMDL	reagent method detection level
RMEAN	reagent blank mean
SIGC	control standard recovery standard deviation
SIGR	reagent blank standard deviation
SIGS	spike recovery standard deviation
SMEAN	mean spike recovery (%)
SQC	percent standard deviation
T(24)	95% confidence student T array
TTO	the terminal I/O channel
X	dummy real number
XLCT	known control standard amount
XLSA2	known spike amount
XMDL	matrix method detection limit
XQC	percent recovery of control or spike
YMDL	the MDL assessment value

summed in the AMAT array.

Two instances require the use of the external array RESPF. The first involves response factors. It is necessary to preview the list of compounds to identify the internal standards and store the ratio of concentration to area for each. Once the internal standards are surveyed, the pointer is positioned to the first compound in the reduced data file. As each record is read, the area to concentration ratio is computed and multiplied times the corresponding member of the RESPF array, thus providing the response factor for the given compound.

The second use of the RESFP array allows for temporary storage of the original unknown concentrations during replicate statistics analysis of spiked unknowns. This procedure appears complicated until one realizes that the parameter of interest is simply the measured concentration of the spiked sample minus the measured concentration of the original sample (i.e. LSF-LSO). Note that in the this process, a bypass loop is used to skip duplicate sample results if they are present in the reduced data file.

Once summations have been performed for all acceptable samples, the mean or standard deviation is computed with the appropriate equation. When both parameters have been computed for the entire set of compounds, the DMPR2 program chains the DMPR3 program.

Figure 8.2 shows the flow of the DMPR2 program and Table 8.1 identifies the variables used in the program.

8.3 DMPR3 - THE STATISTICS REPORTING PROGRAM

The DMPR3 program generates one of six different types of reports depending on the parameter of interest. The program first opens the QC/MT file and reads the header. Next it opens the replicate report file (Section

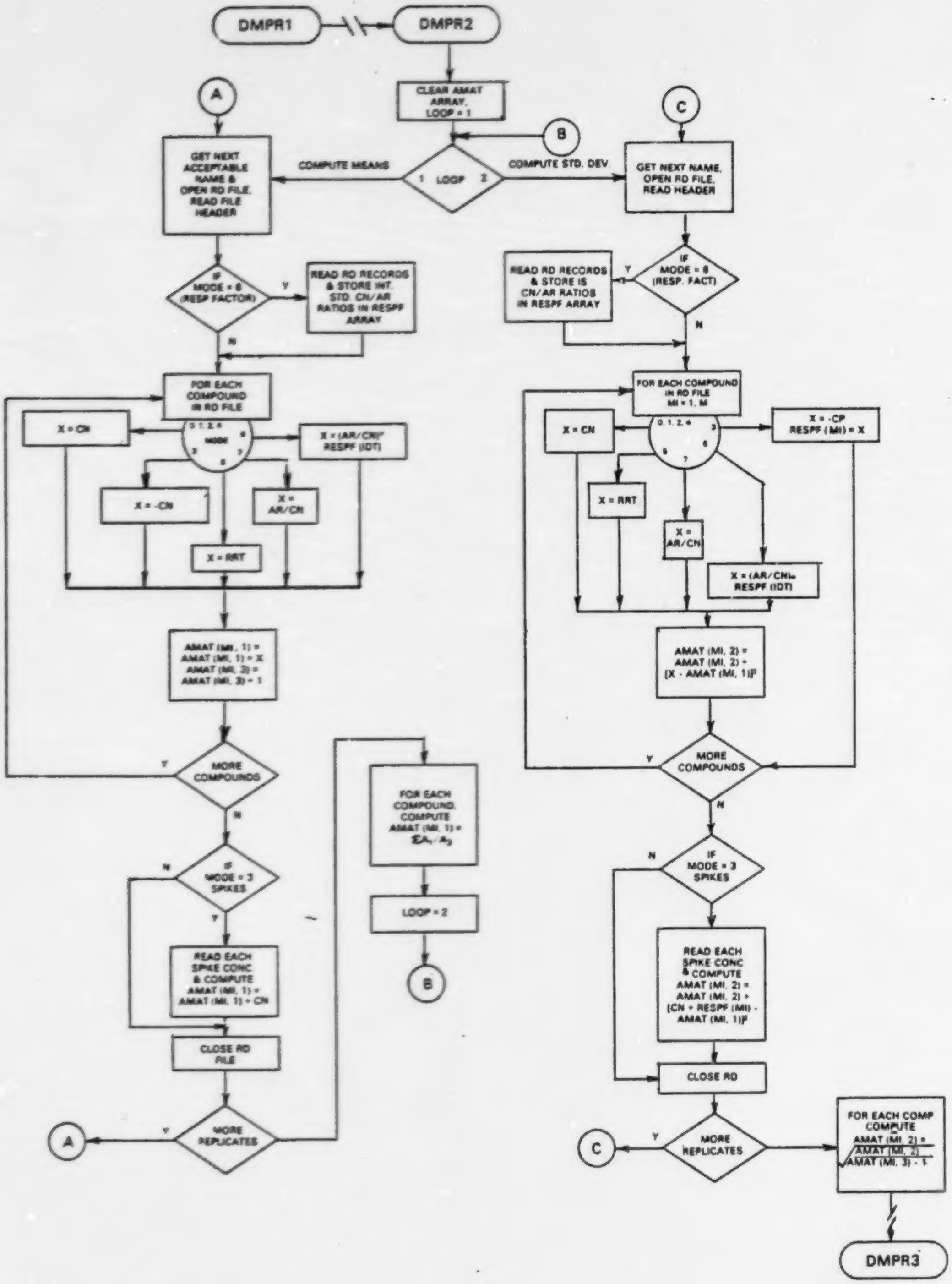

DMPR1
DMPR2
CLEAR AMAT ARRAY, LOOP = 1
A
B
1 LOOP 2
COMPUTE MEANS
COMPUTE STD. DEV.
GET NEXT ACCEPTABLE NAME & OPEN RD FILE. READ FILE HEADER
IF MODE = 6 (RESP FACTOR)
Y
READ RD RECORDS & STORE INT. STD. CN/AR RATIOS IN RESPF ARRAY
N
FOR EACH COMPOUND IN RD FILE
0, 1, 2, 4
MODE
3
5
6
7
X = CN
X = (AR/CN)* RESPF (IDT)
X = -CN
X = AR/CN
X = RRT
AMAT (MI, 1) = AMAT (MI, 1) + X AMAT (MI, 3) = AMAT (MI, 3) + 1
Y
MORE COMPOUNDS
N
IF MODE = 3 SPIKES
N
Y
READ EACH SPIKE CONC & COMPUTE AMAT (MI, 1) = AMAT (MI, 1) + CN
CLOSE RD FILE
A
Y
MORE REPLICATES
FOR EACH COMPOUND, COMPUTE AMAT (MI, 1) = ΣA₁ / A₃
LOOP = 2
B
C
GET NEXT NAME, OPEN RD FILE, READ HEADER
IF MODE = 6 (RESP. FACT)
Y
READ RD RECORDS & STORE IS CN/AR RATIOS IN RESPF ARRAY
N
FOR EACH COMPOUND IN RD FILE MI = 1, M
0, 1, 2, 4
3
5
6
7
X = CN
X = -CP RESPF (MI) = X
X = RRT
X = AR/CN
X = (AR/CN)* RESPF (IDT)
AMAT (MI, 2) = AMAT (MI, 2) + [X - AMAT (MI, 1)]²
Y
MORE COMPOUNDS
N
N
IF MODE = 3 SPIKES
READ EACH SPIKE CONC & COMPUTE AMAT (MI, 2) = AMAT (MI, 2) + [CN + RESPF (MI) - AMAT (MI, 1)]²
CLOSE RD
C
Y
MORE REPLICATES
FOR EACH COMP COMPUTE AMAT (MI, 2) = √(AMAT (MI, 2) / AMAT (MI, 3) - 1)
DMPR3

Figure 8.2 Flow Chart of the DMPR2 Program

11.8) in the write mode and outputs the report header to it. The acceptable sample names are then output to the report file.

The generation of the report subheader is complicated by the number of different types of reports which can be generated. Each of three subheader lines is ultimately composed of from one to three strings identified in the KH arrays.

In the generation of the main body of the report, the program loops through each compound in the list. It reads a record from the QC/MT and outputs the statistics held in the AMAT array. An additional computation is made to obtain the relative standard deviation.

A special algorithm has been included for the cases in which control standard and normal spiked samples are handled. The average percent recovery and the percent standard deviation are computed for quality control charting purposes. These values will be used to update spike and control standard limits in the DMPR4 program.

Another special algorithm is applied to control standards and matrix spikes (adulterated non-SFC samples) in order to obtain and check method determination levels (MDLR and MDLX). This is done by multiplying the standard deviation by the student-t value. (These values are held in the T-array and represent the 95% confidence level.) If the mean measured value is less than FMDL times the computed MDL, then a plus sign (+) is output next to the computed MDL to indicate the acceptablity of the value. If the mean measured value is greater than FMDL times the computed MDL, then the result is flagged with a minus sign (-). The value of FMDL is set in the code at the beginning of the DMPR1 program. Its value has been tentatively set to 5.0. Further testing may require that the number be change or even that it be input by the analyst at the start of a replicate statistics session.

Once all of the compounds have been dealt with, the files are closed and the report file is copied to the line printer. The DMPR3 program then chains to the DMPR4 program.

Figure 8.3 shows the flow of the DMPR3 program and Table 8.1 identifies the variables used in the program.

8.4 DMPR4 - THE QC/MT STATISTICAL UPDATE PROGRAM

The DMPR4 program allows the user to update certain parameters in the QC/MT file with the values obtained in the replicate statistics module. The only fields that can be affected in this way are the reagent MDL, the matrix MDL, the control standard limits, and the normal spike limits. Both reagent MDL and control standard limits can be updated with a single set of control standard replicates.

The DMPR4 program first checks the sample code and parameter of interest to establish that one of the four updates is possible. If not, it immediately chains to the DMPC program. If the template can be updated, the program will prompt the analyst as to whether it is desirable to do so. In the case of control standards, the analyst is allowed to select which parameters are to be updated. The program then opens the QC/MT file in the modify mode and proceeds to update each record. The only instance in which the parameters are not modified is when the expected control standard or spike amounts in the QC/MT file are less than or equal to zero. During the update process, a modification report (DMPRPTM.99) is generated and at the completion of the update, the file is copied to the lineprinter. The report identifies what the old parameter was and what the new one is. At the

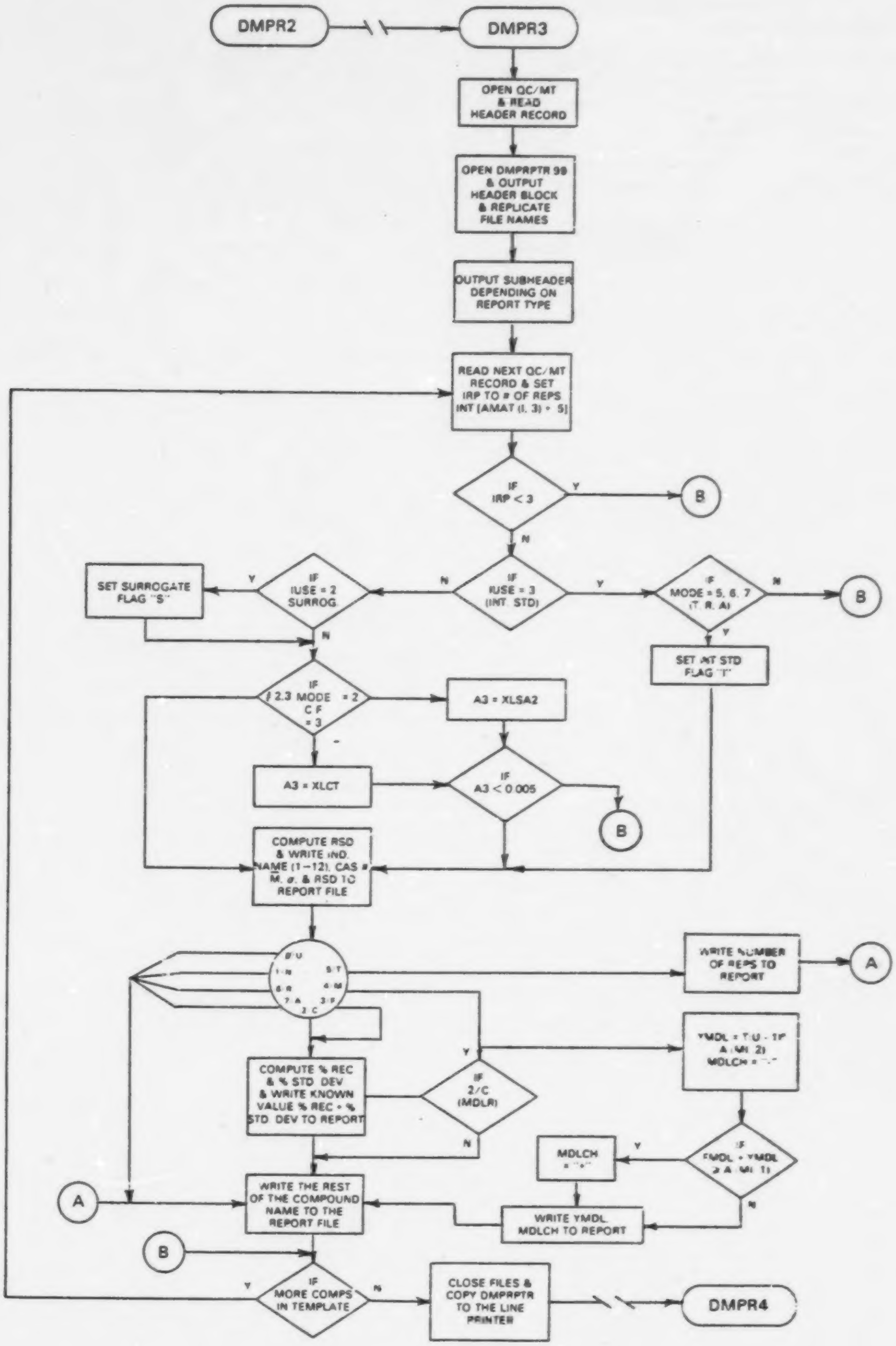

DMPR2
DMPR3
OPEN QC/MT & READ HEADER RECORD
OPEN DMPRPTR 99 & OUTPUT HEADER BLOCK & REPLICATE FILE NAMES
OUTPUT SUBHEADER DEPENDING ON REPORT TYPE
READ NEXT QC/MT RECORD & SET IRP TO # OF REPS INT [AMAT (I, 3) + .5]
IF IRP < 3
Y
B
N
IF IUSE = 3 (INT. STD)
N
IF IUSE = 2 SURROG.
Y
SET SURROGATE FLAG "S"
N
Y
IF MODE = 5, 6, 7 (T, R, A)
N
B
Y
SET INT STD FLAG "I"
IF 2.3 MODE = 2 C F = 3
A3 = XLSA2
A3 = XLCT
IF A3 < 0.005
B
COMPUTE RSD & WRITE IND. NAME (1–12), CAS # M, σ, & RSD TO REPORT FILE
8/U
1/N
5/T
6/R
4/M
7/A
3/F
2/C
WRITE NUMBER OF REPS TO REPORT
A
Y
YMDL = T(U - 1F A (M, 2) MDLCH = " "
COMPUTE % REC & % STD. DEV & WRITE KNOWN VALUE % REC + % STD. DEV TO REPORT
IF 2/C (MDLR)
N
IF FMDL + YMDL ≥ A (M, 1)
Y
MDLCH = "+"
N
A
WRITE THE REST OF THE COMPOUND NAME TO THE REPORT FILE
WRITE YMDL, MDLCH TO REPORT
B
Y
IF MORE COMPS IN TEMPLATE
N
CLOSE FILES & COPY DMPRPTR TO THE LINE PRINTER
DMPR4

Figure 8.3 Flow Chart of the DMPR3 Program

completion of the update, the DMPR4 program chains the DMPC program. The DMPR4 program has not been implemented.

SECTION 9

DMPP - THE SAMPLE FILE CONTROL RUN RESULTS PRODUCTION PROGRAMS

The DMPP module packages SFC compatible run results files (Section 11.9). The module is composed of two programs. The DMPP1 program collects sample data from reduced data files (Section 11.5) and produces a run results file. The DMPP2 program optionally appends local quality control limits to the file and then transmits the file either to the microprocessor buffer interface or to magnetic tape. An option in the DMPP2 program has been set aside for the future interface through the modem port to the PDP 11/70. A special assembly language subroutine will be required before this option can be implemented.

9.1 DMPP1 - THE RUN RESULTS FILE GENERATION PROGRAM

The DMPP1 program first gets the date and time and constructs the run results file name of the form YYMMDDHHmm.RR. It then prompts the analyst for the precise PARMETH code for the results contained in the file. Next the type-1 header record is constructed, the run results file is opened in the write mode, and the header is written to the results file.

The analyst is then prompted for the mode of entering file names. The list mode (L), the manual entry mode (M), or the session termination mode (X) are the available options. Any number of lists or manually entered file names may be used. The program will always return to this prompt line until the exist option (X) is selected.

If a name list is chosen, the program certifies its existence and then opens and reads it. It then cycles through each name and accesses results from locatable reduced data files.

For each legitimate reduced data file, the DMPP1 program opens the file and reads the header record. It constructs the associated QC/MT file name and opens the template in the read mode. The template header is then read.

For each unique occurrence of sample type and QC/MT name, the program updates the IQCLS array. This array is used in the DMPP2 program to limit the number of quality control records appended to the run results file. IQCLS array is searched for each reduced data file processed by the program. If a match is not found, the number of items (IQCNM) is incremented by one and the new member is assigned to the group. (See Section 9.2 for further details.)

A brief description of the data being processed is output to the terminal and the program generates a type-6 run results subheader record. It then processes the reduced data file by reading a record from the QC/MT file and then reading a record from the reduced data file. Depending on the type of sample being processed, the program transfers control to the appropriate routine.

For each record written in the results file, counters are incremented to account for the number and kind of records written. Count fields are used to locate subheader records and to track the number of compounds reported. (Internal standards are ignored completely.) When a reduced data file has been processed, the DMPP1 program closes the QC/MT file and positions the reduced data file pointer to the 149th byte in the header. The run results file name and data are then written into the header and the file is closed.

If the reduced data file contains unknown information (U), additional processing is performed. The program first sets up the string identifier for the primary sample on the basis of the sample code (i.e., bbb,LSO, LD1, or LDX). After processing all of the compounds in the primary segment, it changes the descriptor field, updates the IQCLS list and generates a new type-6 subheader record for either duplicate or spiked sample data. The QC/MT is reset and the header is reread. Both files are then read and the run results records are written and tallied. When unknown processing of quantitative data has been completed, the program examines the reduced data file for qualitative identification data. The program will read and output qualitative run results until the end of the reduced data is reached. Special type-7 records are written containing the names of each compound identified. If an actual concentration is available, its value is obtained and included in the field of the type 2 record. In this case, the "present but not quantified" (PNQ) qualifier is masked.

Processing continues until the analyst has no more files to include in the run results file. At that point, the DMPP1 program closes all files and reopens the run results file in the modify mode. It then positions to the count field in each type-6 record and writes the corresponding integer count into the field. When it has completed this task, the DMPP1 program chains to the DMPP2 program.

Figures 9.1a and 9.1b show the flow of the DMPP1 program and Table 9.1 identifies the variables used in the program.

9.2 DMPP2 - THE RUN RESULTS FILE TRANSMISSION PROGRAM

The DMPP2 program performs two distinct functions. The first is optional appending of QC limits to the run results file. The second is run results transmission to the SFC system.

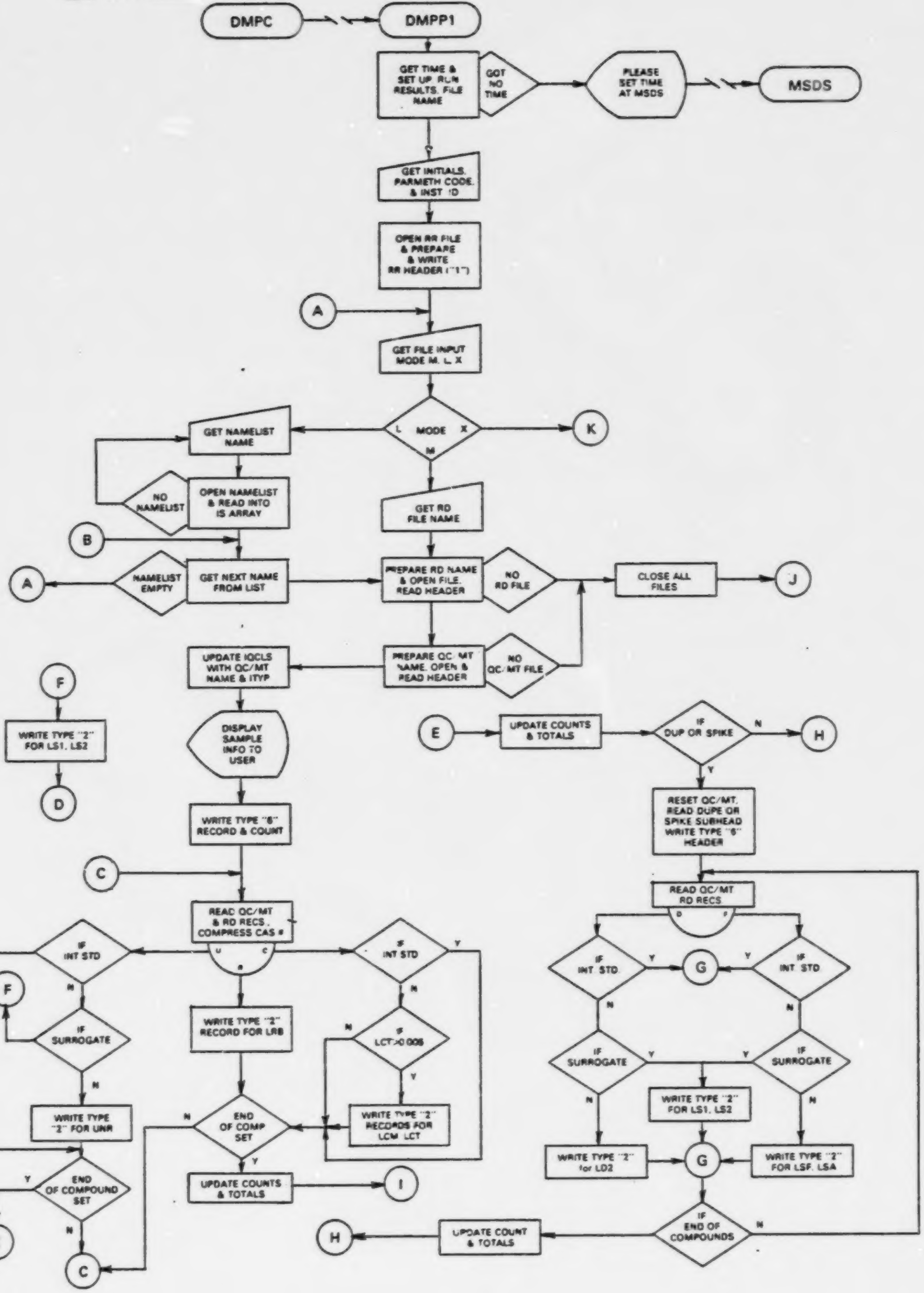

DMPC
DMPP1
GET TIME & SET UP RUN RESULTS. FILE NAME
GOT NO TIME
PLEASE SET TIME AT MSDS
MSDS
GET INITIALS, PARMETH CODE, & INST ID
OPEN RR FILE & PREPARE & WRITE RR HEADER ("1")
A
GET FILE INPUT MODE M, L, X
MODE
L
X
M
K
GET NAMELIST NAME
NO NAMELIST
OPEN NAMELIST & READ INTO IS ARRAY
GET RD FILE NAME
B
A
NAMELIST EMPTY
GET NEXT NAME FROM LIST
PREPARE RD NAME & OPEN FILE. READ HEADER
NO RD FILE
CLOSE ALL FILES
J
UPDATE IOCLS WITH QC/MT NAME & ITYP
PREPARE QC/MT NAME. OPEN & READ HEADER
NO QC/MT FILE
F
WRITE TYPE "2" FOR LS1, LS2
D
DISPLAY SAMPLE INFO TO USER
WRITE TYPE "6" RECORD & COUNT
E
UPDATE COUNTS & TOTALS
IF DUP OR SPIKE
N
H
Y
RESET QC/MT, READ DUPE OR SPIKE SUBHEAD WRITE TYPE "6" HEADER
C
READ QC/MT & RD RECS, COMPRESS CAS #
U
C
R
IF INT STD
N
F
IF SURROGATE
N
WRITE TYPE "2" FOR UNR
D
END OF COMPOUND SET
Y
E
N
C
WRITE TYPE "2" RECORD FOR LRB
END OF COMP SET
N
Y
UPDATE COUNTS & TOTALS
I
IF INT STD
Y
N
IF LCT>0.005
N
Y
WRITE TYPE "2" RECORDS FOR LCM LCT
READ QC/MT RD RECS
D
F
IF INT STD
Y
G
Y
IF INT STD
N
N
IF SURROGATE
Y
Y
IF SURROGATE
WRITE TYPE "2" FOR LS1, LS2
N
N
WRITE TYPE "2" for LD2
G
WRITE TYPE "2" FOR LSF, LSA
IF END OF COMPOUNDS
N
UPDATE COUNT & TOTALS
H

Figure 9.1a Flow Chart of the DMPP1 Program

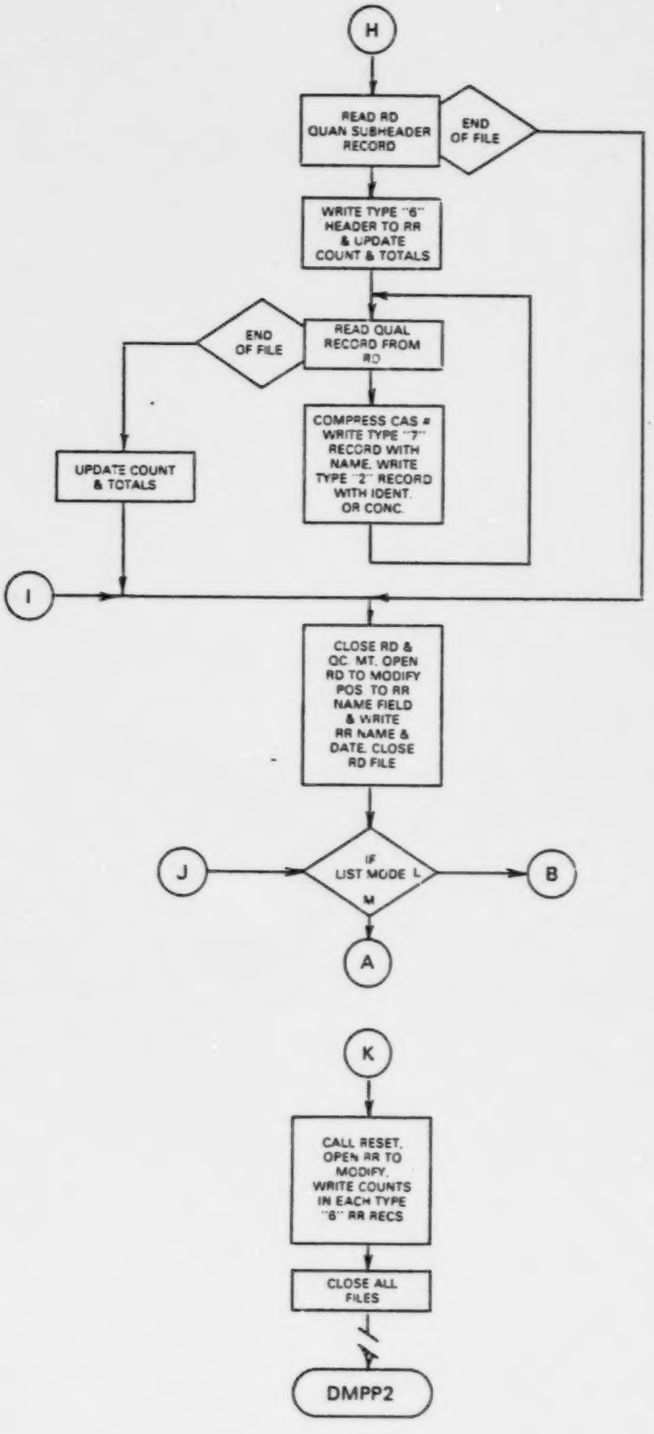

H
READ RD QUAN SUBHEADER RECORD
END OF FILE
WRITE TYPE "6" HEADER TO RR & UPDATE COUNT & TOTALS
END OF FILE
READ QUAL RECORD FROM RD
COMPRESS CAS # WRITE TYPE "7" RECORD WITH NAME. WRITE TYPE "2" RECORD WITH IDENT. OR CONC.
UPDATE COUNT & TOTALS
I
CLOSE RD & QC. MT. OPEN RD TO MODIFY POS. TO RR NAME FIELD & WRITE RR NAME & DATE. CLOSE RD FILE
J
IF LIST MODE
L
M
B
A
K
CALL RESET. OPEN RR TO MODIFY. WRITE COUNTS IN EACH TYPE "6" RR RECS
CLOSE ALL FILES
DMPP2

Figure 9.1b Flow Chart of the DMPP1 Program

TABLE 9.1

VARIABLES DESCRIPTIONS FOR THE DMPP PROGRAMS

VARIABLE	DESCRIPTION
A	method range minimum
AR	area
B	method range maximum
C	duplicate limit - slope
CMEAN	mean control standard recovery (%)
CONC	concentration
D	duplicate limit - intercept
D1 - D4	rounding constants
FACTOR	concentration multiplier
I	counter
I1	string position pointer
I2	string position pointer
I6CNT	type '2' record counter
I6CNT	type '6' record counter
I7CNT	type '7' record counter
IACCNT(4)	account code or SFC sample identifier
IACDT(4)	acquisition date for primary sample
IACDT2(4)	acquisition date for spike or duplicate
IANLST(4)	analyst name or initials
IANS	answer string
IC	number of names in namelist
ICAS(6)	compound CAS number string
ICCNT	control standard compounds reported
ICD	duplicate or spike code
ICH	file input channel number
ICHK	ICODE check digit
ICKST(5)	name of reference control standard
ICMNT(32)	DMP user comment
ICODE	sample type code
ICOND(32)	comment on measurement conditions
ICOUNT(*,1)	the number of result items reported
ICOUNT(*,2)	locations of type '6' records in run results file
ICP	sample code counter for spikes and duplicates
ID	qualitative identification process code
IDAY	day
IDT	internal standard reference
IE	error code
IEE	I/O error variable
IER	error code
IEXDT(4)	extraction date for primary sample
IEXDT2(4)	extraction date for duplicate or spike

TABLE 9.1 (Continued)

VARIABLE	DESCRIPTION
IEXTC(2)	QC/MT filename extension
IEXTR(2)	reduced data file extension
IFCNT	spike compounds reported
IFLDSC(11)	general DMP file descriptor
IFLFN(10)	general DMP file specifier
IFLNM(8)	general DMP file name
IFORM(10)	qualitative identification compound formula
IHR	hour
IMIN	minute
IMO	month
INDQ	QC/MT index
INIT(2)	analyst's initials
INLDSC(11)	namelist file descriptor
INLNM(8)	namelist file name
INST(3)	instrument name
IPMCT	reduced data file compound count
IPMTH(4)	fixed list PARMETH code
IPRDT(4)	process date
IPVTH(4)	variable list PARMETH code
LQADT(4)	qualitative identification process date
IQBKS	number of qualitative identification compounds reported
IQCC(12)	SFC sample type super string
IQCDSC(11)	QC/MT filename descriptor
IQCFN(8)	QC/MT filename specifier
IQCLS(10,*)	list of PARMETH/sample type pairs for limit appending
IQCNM(4)	QC/MT name (PARMETH)
IQCT1(2)	primary SFC code (i.e., LD1, LSO, LDX)
IQCT2(2)	duplicate or spike SFC code (i.e., LD2, LSF/LSA)
IQCT3(2)	surrogate QC code for measured value (LS1)
IQCT4(2)	surrogate QC code for known value (LS2)
IQLNM	number of PARMETH/sample type pairs for limit appending
IRCAS95)	compressed CAS number string
IRGBK95)	name of reference reagent blank
IRRCH(9)	dummy run results file name
IRRDT(4)	run result formation date
IRRNM(9)	SFC run result file name
IRRT	integerized relative retention time
IS(11,MAXSM)	namelist name array
ISAMP	namelist sample counter
ISCNT	surrogate compounds recorded
ISEC	second
ISMPLE(32)	comment on sample
ISMPLS	number of samples reported
ISPEC	qualitative identification spectral number
ISTOP(4)	stopping characters for NOVA transmission
ISTRT94)	start characters for NOVA transmission
ITMP(20)	a text string
ITOTAL	run results record counter

TABLE 9.1 (continued)

VARIABLE	DESCRIPTION
ITXT(50)	a text string
ITYPE	duplicate/spike character code
ISUE	compound use code
IX	dummy integer
IYR	year
KEYC	source of control standard limits
KEYD	source of duplicate limits
KEYR	source of reagent blank limits
KEYS	source of spike limits
KUNITS(3)	the units of measure
KVOL(2)	sample volume indicator
LCH	file I/O channel number
LN	length of sample name (location of type code)
LNN	LN+1
LPT	the lineprinter channel
LSTMD	list mode
MCH	report channel
MCODE(4)	sample type super string
METH(2)	peak measurement mode
MW	qualitative identification molecular weight
NAME(36)	the compound name string
NCOMP(20	computation mode indicator
NIND	the number of compounds
NM	number of compounds in QC/MT
NN	number of bytes per QC/MT record
NO	QC/MT header offset (bytes)
NPIND	duplicate or spike compound index
NSAMP	number of samples in namelist
RMDL	reagent method detection level
RMEAN	reagent blank mean
SIGC	control standard recovery standard deviation
DIGR	reagent blank standard deviation
SIGS	spike recovery standard deviation
SMEAN	mean spike recovery (%)
TTO	the terminal I/O channel
VEX	extract volume
VINJ	injection volume
VOL	sample volume
X1 - X7	rounded report variables
XLCT	known control standard amount
XLSA2	known spike amount
XMDL	matrix method detection level

The DMPP2 program has been made independent of DMPP1 so that any ASCII file can be transmitted to the host minicomputer. The DMPP2 program first obtains the date and time from the system clock and checks it against the expected run results file name (held in COMMON). If the program has been accessed from somewhere other than DMPP1, this check will fail. Once the program determines that it has been accessed independently, it prompts the analyst for a full file name and after checking its existence, it transfers control to the file transmission segment of the program.

If the run results file name is compatible with the date/time information, the program branches to the limit appending routine. The analyst is prompted as to whether or not limits are to be appended to the file. If the choice is to append limits, the run results file is opened in the modify mode and the pointer is positioned to the end of the file. The program then loops through the QC/MT names and sample types stored in the IQCLS array. For each item, the program opens the template and reads the header. The program then loops through each compound in the template and generates the appropriate limit records. Once limit records have been appended for each distinct quality control sample type, the program branches to the file transfer segment of the program.

The file transmission segment of the DMPP1 program is option oriented and facilitates the output of an 80-byte ASCII file to either the magnetic tape (T), the NOVA 840 buffer interface (N), or the PDP 11/70 buffer interface (P). If the file is to be transferred to magnetic tape, the analyst is prompted to specify the file number on the tape where the file is to be placed. The selected mag tape file is then opened in the write mode and the ASCII file is copied directly to it. Extensive error processing has been coded into this segment.

The option to transfer the file to the NOVA 840 (N) is uncomplicated. The interface hardware is directly coupled to the user terminal so the analyst is instructed to turn on the interface and press the return button. The start characters are sent, then the file is copied to the terminal, and the stop characters are sent. The analyst is then instructed to turn off the interface.

The option to transfer the file to the PDP 11/70 (P) is accomplished through the use of assembler language calls which prepare the modem channel and send individual records to the interface buffer. In this case, each ASCII file record is read and transmitted until an end of file is reached. Hand shaking is done internally to the call. If record reception is not acknowledged, the program repeats the record. If the number of transmission failures exceeds MAXERR, the session is scratched and an error message is generated.

When transmission of the file has been completed, the analyst is given the opportunity to output the ASCII file to the line printer and to delete the file from disk. When these requests have been serviced, the program chains to the DMPC program.

Figure 9.2 shows the flow of the DMPP2 program and Table 9.1 describes the variables used in the program.

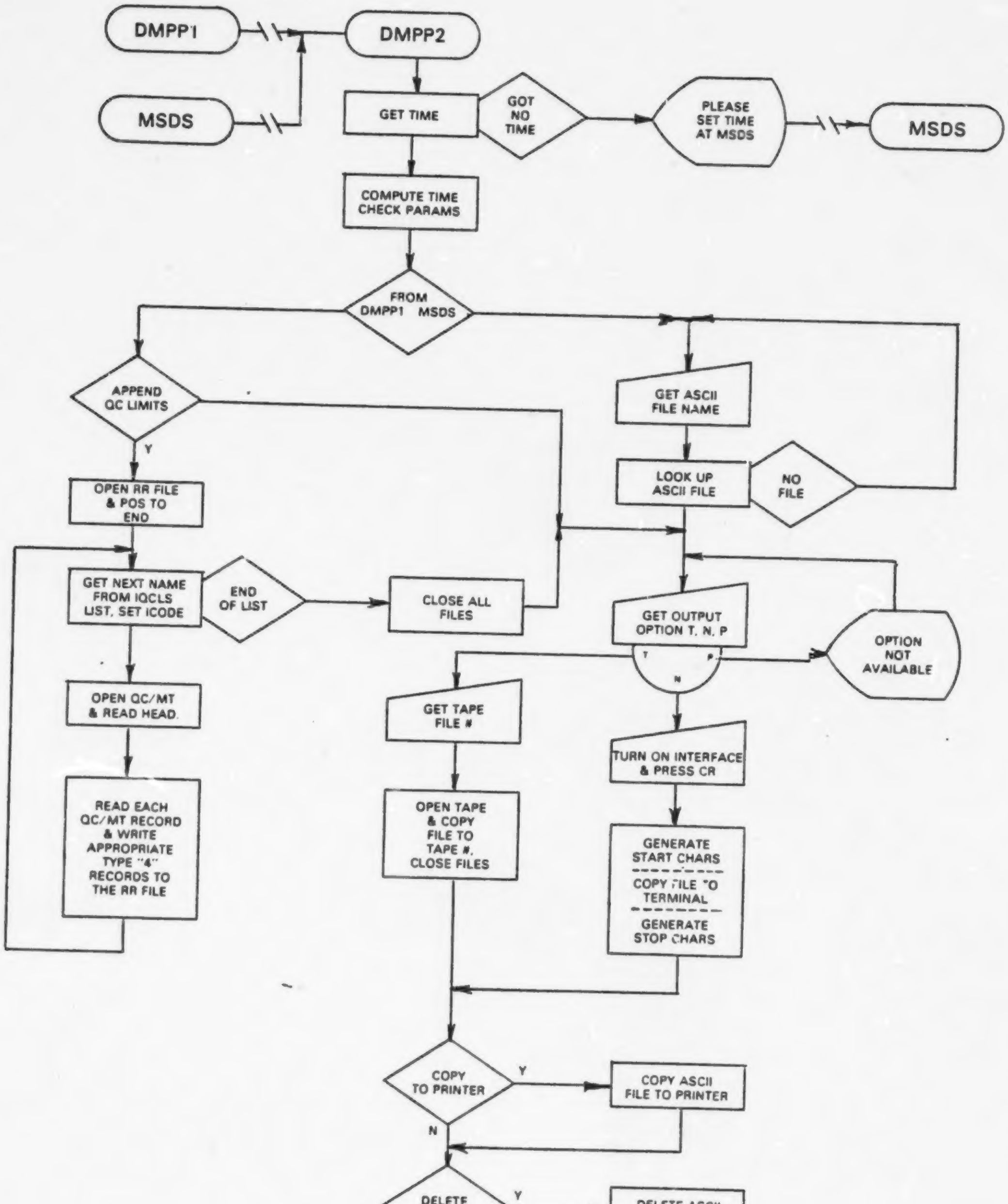

DMPP1
MSDS
DMPP2
GET TIME
GOT NO TIME
PLEASE SET TIME AT MSDS
MSDS
COMPUTE TIME CHECK PARAMS
FROM DMPP1 MSDS
APPEND QC LIMITS
Y
OPEN RR FILE & POS TO END
GET NEXT NAME FROM IQCLS LIST, SET ICODE
END OF LIST
CLOSE ALL FILES
OPEN QC/MT & READ HEAD.
READ EACH QC/MT RECORD & WRITE APPROPRIATE TYPE "4" RECORDS TO THE RR FILE
GET ASCII FILE NAME
LOOK UP ASCII FILE
NO FILE
GET OUTPUT OPTION T, N, P
T
N
P
OPTION NOT AVAILABLE
GET TAPE FILE #
OPEN TAPE & COPY FILE TO TAPE #, CLOSE FILES
TURN ON INTERFACE & PRESS CR
GENERATE START CHARS
COPY FILE TO TERMINAL
GENERATE STOP CHARS
COPY TO PRINTER
Y
N
COPY ASCII FILE TO PRINTER
DELETE FILE
Y
DELETE ASCII FILE
DMPC

Figure 9.2 Flow Chart of the DMPP2 Program

SECTION 10
DMPU - THE DMP UTILITY PROGRAM

The DMPU program is a utility program which performs simple functions through an option oriented command line. These functions include the direct access of DMPG2 for reduced data file reporting, the direct access of DMPP2 for transmission of ASCII files to other computers, the output of extant report files to the lineprinter, and the archiving of sample data files to magnetic tape.

The DMPU program has not been implemented.

SECTION 11
DMP DATA AND REPORT FILES

The DMP system interacts with a number of disk files to perform its functions. The system reads sample quantitation files and qualitative identification files. It writes to and reads from sample log files. The system creates reduced data files ($.RD) and fully maintains the quality control method template files (*.QC). The DMP system produces three primary report files. It also produces SFC compatible run results files. The DMP system also reads SFC supplied quality control limit files. This section will describe each of these files in terms of content and form.

11.1 $.LG - THE SAMPLE LOG FILE

The sample log file ($.LG) is created by the INCOS system at sample measurement time. It initially contains information on the acquisition of data for a sample. After a sample has been run, the analyst uses the DMPL program to enter volumetric and data base management parameters at the end of the log file. The DMPG1 program accesses the log file header information and the volumetric data in the reduced data file creation process. The format of the header segment of the log file is shown in Figure 11.1a. The format of the volumetric and data base management parameters are shown in Figure 11.1b.

11.2 $.QN - THE SAMPLE QUANTITATION FILE

The quantitation file is simply a renamed QUAN.99 file. It is copied to the $.QN name by the DMPN program which normally operates in a user

```
MM/DD/YY    HH:MM:SS          SCAN 1 OF NNNN
     ACQUISITION STARTED

ACQUIRE                     RUN D:FILENAME           ACQUIRING
MM/DD/YY HH:MM:SS +  0:01   FREE SECTORS: NNNN       SCAN: 1 OF NNNN
SAMPLE: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
CONDS.: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
FORMULA:                    INSTRUMENT: INSTNM       WEIGHT:    0.000
SUBMITTED BY: @@@@@@        ANALYST: @@@@@@          ACCT. NO.: NNNNNNNN
        .              .                       .              .
        .              .                       .              .
        .              .                       .              .
        .              .                       .              .
```

FIGURE 11.1A LOG FILE HEADER RECORD FORMAT

```
        .               .              .              .
        .               .              .              .
        .               .              .              .
        .               .              .              .
        .               .              .              .

 6/27/83    11:37:15
     ACQUISITION COMPLETED
     SCANS    1 TO    2 CENTROID
     SCANS    3 TO  123 SUSPENDED
     SCANS  124 TO 1930 CENTROID

MODE      SCANS    SECS  OUT OF       %     PEAKS PER SCAN  PER SEC
CENTROID   1809   254.3  1809.0    14.1    255271.     141.     141.

07/13/83  14:47:45
$$$
VS= #####.###  VE=  #####.###  XD= 7/13/83   VI=  #####.###
PM= XXXYZMN   CS= CHKSTDNAME  RB= REAGBLNAME  PV= XXXYZLN
CM= @@@@@@@@@@@@@@@@@@@@@@@@3@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

FIGURE 11.1B LOG FILE DATA MANAGEMENT AND VOLUMETRIC PARAMETERS

procedure. The $.QN file may be read by the DMPG1 and the DMPQ1 programs. The $.QN file has numerous forms which can be set through the QUAN command. The essential information includes a list of compounds names, a list of quantitative data, and for standards, a list of response data . An example of the QUAN report is shown in Figure 11.2.

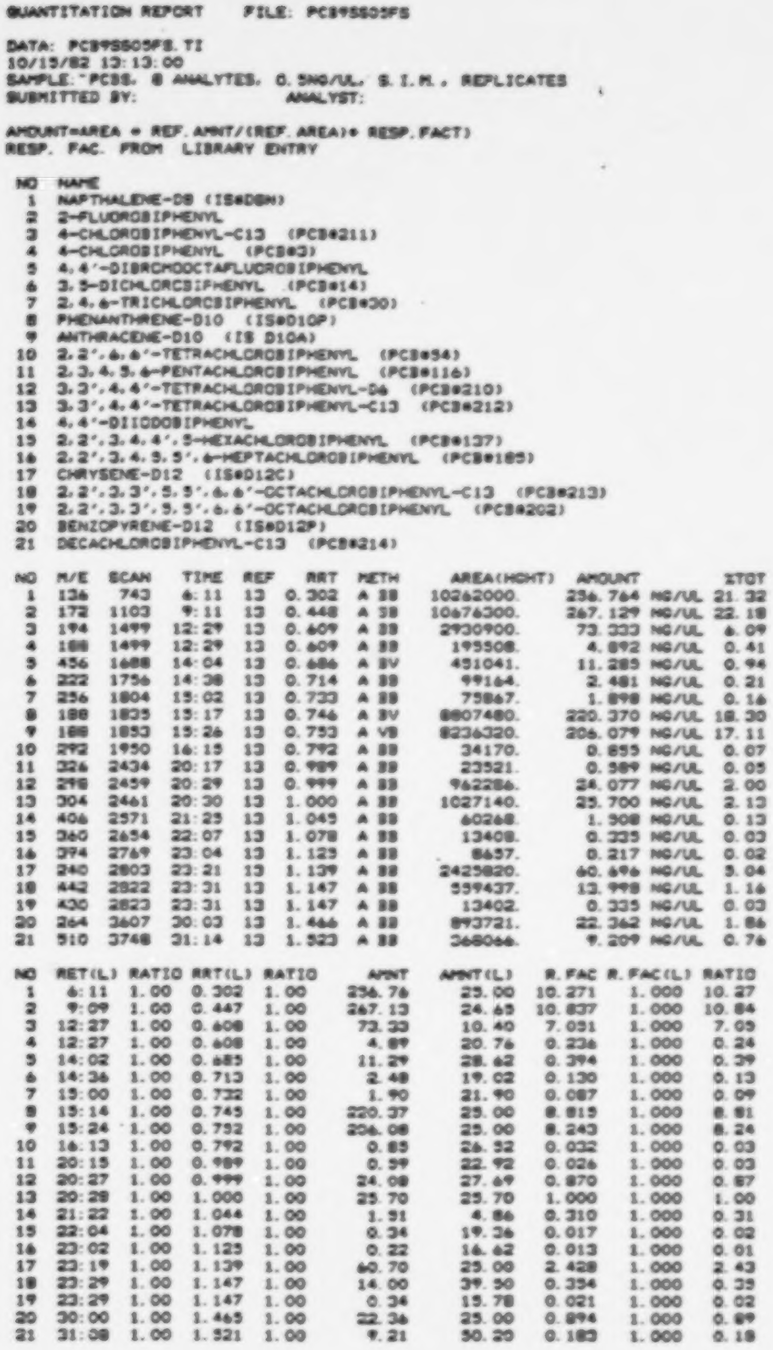
11.3 $.QA - THE SAMPLE QUALITATIVE IDENTIFICATION FILE

The qualitative identification file contains information on compounds detected in an environmental sample but not quantified through the method. The QUALT program produces a list of prospects ($.QZ) during the library search process. The analyst then uses the QUAL procedure to select the best matches for identified compounds. The results of the QUAL session is the qualitative identification file ($.QA). The contents of this file are shown in Figure 11.3. The DMPG1 looks for the $.QA file and appends its contents to the reduced data file (RD).

11.4 PARMETH.QC - THE QUALITY CONTROL METHOD TEMPLATE FILE

The quality control method template file (QC/MT) is an essential feature of the DMP system. Any number of template files can be maintained through the DMPQ module. Each template file contains the names and CAS numbers of the compounds measured for a given method. It also contains compound use codes which identify internal standards and surrogate spikes. Finally, the template contains method data on detection limits and quality control assessment criteria for each compound in the fixed list. Each set of assessment limits is followed by a limit key code. These codes show where the limits came from. The contents of the QC/MT is shown in Table 11.4.

A QC/MT file is accessed by nearly every DMP program. Since record I/O is performed on the file, the header contains the header offset size, the



```
QUANTITATION REPORT    FILE: PCB95505FS

DATA: PCB95505FS.TI
10/15/82 13:13:00
SAMPLE: PCBS, 8 ANALYTES, 0.5NG/UL, S.I.M., REPLICATES
SUBMITTED BY:                 ANALYST:

AMOUNT=AREA * REF.AMNT/(REF.AREA)* RESP.FACT)
RESP. FAC. FROM  LIBRARY ENTRY

 NO  NAME
  1  NAPTHALENE-D8 (IS#D8N)
  2  2-FLUOROBIPHENYL
  3  4-CHLOROBIPHENYL-C13  (PCB#211)
  4  4-CHLOROBIPHENYL  (PCB#3)
  5  4,4'-DIBROMOOCTAFLUOROBIPHENYL
  6  3,5-DICHLOROBIPHENYL  (PCB#14)
  7  2,4,6-TRICHLOROBIPHENYL  (PCB#30)
  8  PHENANTHRENE-D10  (IS#D10P)
  9  ANTHRACENE-D10  (IS D10A)
 10  2,2',6,6'-TETRACHLOROBIPHENYL  (PCB#54)
 11  2,3,4,5,6-PENTACHLOROBIPHENYL  (PCB#116)
 12  3,3',4,4'-TETRACHLOROBIPHENYL-D6  (PCB#210)
 13  3,3',4,4'-TETRACHLOROBIPHENYL-C13  (PCB#212)
 14  4,4'-DIIODOBIPHENYL
 15  2,2',3,4,4',5-HEXACHLOROBIPHENYL  (PCB#137)
 16  2,2',3,4,5,5',6-HEPTACHLOROBIPHENYL  (PCB#185)
 17  CHRYSENE-D12  (IS#D12C)
 18  2,2',3,3',5,5',6,6'-OCTACHLOROBIPHENYL-C13  (PCB#213)
 19  2,2',3,3',5,5',6,6'-OCTACHLOROBIPHENYL  (PCB#202)
 20  BENZOPYRENE-D12  (IS#D12P)
 21  DECACHLOROBIPHENYL-C13  (PCB#214)

 NO  M/E  SCAN   TIME  REF   RRT  METH    AREA(HGHT)  AMOUNT          %TOT
  1  136   743   6:11   13  0.302  A BB   10262000.   256.764 NG/UL 21.32
  2  172  1103   9:11   13  0.448  A BB   10676300.   267.129 NG/UL 22.18
  3  194  1499  12:29   13  0.609  A BB    2930900.    73.333 NG/UL  6.09
  4  188  1499  12:29   13  0.609  A BB     195508.     4.892 NG/UL  0.41
  5  456  1688  14:04   13  0.686  A BV     451041.    11.285 NG/UL  0.94
  6  222  1756  14:38   13  0.714  A BB      99164.     2.481 NG/UL  0.21
  7  256  1804  15:02   13  0.733  A BB      75867.     1.898 NG/UL  0.16
  8  188  1835  15:17   13  0.746  A BV    8807480.   220.370 NG/UL 18.30
  9  188  1853  15:26   13  0.753  A VB    8236320.   206.079 NG/UL 17.11
 10  292  1950  16:15   13  0.792  A BB      34170.     0.855 NG/UL  0.07
 11  326  2434  20:17   13  0.989  A BB      23521.     0.589 NG/UL  0.05
 12  298  2459  20:29   13  0.999  A BB     962286.    24.077 NG/UL  2.00
 13  304  2461  20:30   13  1.000  A BB    1027140.    25.700 NG/UL  2.13
 14  406  2571  21:25   13  1.045  A BB      60268.     1.508 NG/UL  0.13
 15  360  2654  22:07   13  1.078  A BB      13408.     0.335 NG/UL  0.03
 16  394  2769  23:04   13  1.123  A BB       8657.     0.217 NG/UL  0.02
 17  240  2803  23:21   13  1.139  A BB    2425820.    60.696 NG/UL  5.04
 18  442  2822  23:31   13  1.147  A BB     559437.    13.998 NG/UL  1.16
 19  430  2823  23:31   13  1.147  A BB      13402.     0.335 NG/UL  0.03
 20  264  3607  30:03   13  1.466  A BB     893721.    22.362 NG/UL  1.86
 21  510  3748  31:14   13  1.523  A BB     368066.     9.209 NG/UL  0.76

 NO  RET(L) RATIO RRT(L) RATIO      AMNT    AMNT(L)    R.FAC R.FAC(L) RATIO
  1   6:11  1.00  0.302  1.00     256.76     25.00   10.271    1.000  10.27
  2   9:09  1.00  0.447  1.00     267.13     24.65   10.837    1.000  10.84
  3  12:27  1.00  0.608  1.00      73.33     10.40    7.051    1.000   7.05
  4  12:27  1.00  0.608  1.00       4.89     20.76    0.236    1.000   0.24
  5  14:02  1.00  0.685  1.00      11.29     28.62    0.394    1.000   0.39
  6  14:36  1.00  0.713  1.00       2.48     19.02    0.130    1.000   0.13
  7  15:00  1.00  0.732  1.00       1.90     21.90    0.087    1.000   0.09
  8  15:14  1.00  0.745  1.00     220.37     25.00    8.815    1.000   8.81
  9  15:24  1.00  0.752  1.00     206.08     25.00    8.243    1.000   8.24
 10  16:13  1.00  0.792  1.00       0.85     26.52    0.032    1.000   0.03
 11  20:15  1.00  0.989  1.00       0.59     22.92    0.026    1.000   0.03
 12  20:27  1.00  0.999  1.00      24.08     27.69    0.870    1.000   0.87
 13  20:28  1.00  1.000  1.00      25.70     25.70    1.000    1.000   1.00
 14  21:22  1.00  1.044  1.00       1.51      4.86    0.310    1.000   0.31
 15  22:04  1.00  1.078  1.00       0.34     19.36    0.017    1.000   0.02
 16  23:02  1.00  1.123  1.00       0.22     16.62    0.013    1.000   0.01
 17  23:19  1.00  1.139  1.00      60.70     25.00    2.428    1.000   2.43
 18  23:29  1.00  1.147  1.00      14.00     39.50    0.354    1.000   0.35
 19  23:29  1.00  1.147  1.00       0.34     15.78    0.021    1.000   0.02
 20  30:00  1.00  1.465  1.00      22.36     25.00    0.894    1.000   0.89
 21  31:08  1.00  1.521  1.00       9.21     50.20    0.183    1.000   0.18
```

FIGURE 11.2 EXAMPLE QUAN REPORT PRODUCED BY INCOS SYSTEM

FIELD NUMBER	PARAMETER DESCRIPTION	FORMAT
1	SPECTRUM NUMBER	I5, 1X
2	COMPOUND NAME	34A2/
3	CONCENTRATION (OPTIONAL)	20X, E10.3, 1X
4	CAS NUMBER	6A2, 2X
5	FORMULA	10A2
6	MOLECULAR WEIGHT	I5, 1X
7	IDENTIFICATION CODE	A2/

FIGURE 11.3 RECORD LAYOUT FOR QUALITATIVE RESULTS

TABLE 11.4 QC/METHOD TEMPLATE DESCRIPTION

CONTENTS: KNOWN VALUES AND QC ASSESSMENT CRITERIA.
NAME: ASSIGNED AS XXX0ZNN FROM PARMETH (I.E. 6250202).
MAINTAINED: CREATED, FILLED, AND EDITED BY THE ANALYST THROUGH A TEMPLATE MANAGEMENT ROUTINE (I.E. DMPQ).
LIMITATION: ONE TEMPLATE FOR EACH UNIQUE METHOD DESCRIPTOR (I.E. XXXNN) AND SOURCE DESIGNATOR (I.E. Z).
HEADER: CONTENTS INCLUDE FILENAME (4A2), UNITS (3A2), NUMBER OF RECORDS (I4), HEADER OFFSET (I4), RECORD SIZE (I4,1X), CREATION DATE (4A2,1X), CREATOR (3A2,1X), LAST UPDATE (4A2,1X), PERSON CHANGING FILE (3A2,1X).

FIELD	FIELD CONTENTS	FORMAT
0	COMPOUND NUMBER.	I3,1X
1	COMPOUND NAME	35A2,1X/
2	CHEMICAL ABSTRACT SERVICE (CAS) REGISTRY NUMBER.	6A2,1X
3	INTERNAL STANDARD, SPIKE, OR SURROGATE USE CODE. *	I3,1X
4	REAGENT DETECTION LIMIT IN DEFAULT UNITS (RMDL)	F7.2
4	MATRIX DETECTION LIMIT IN DEFAULT UNITS (XMDL)	F7.2,1X
5	DUPLICATE ASSESSMENT PARAMETER A (MINIMUM)	F10.3
5	DUPLICATE ASSESSMENT PARAMETER B (MAXIMUM)	F10.3
5	DUPLICATE ASSESSMENT PARAMETER C (SLOPE)	F10.3
5	DUPLICATE ASSESSMENT PARAMETER D (INTERCEPT)	F10.3
5	KEYD - DUPLICATE LIMIT SOURCE CODE. **	A1/
6	REAGENT BLANK MEAN RECOVERY IN DEFAULT UNITS.	F7.2
6	REAGENT BLANK STANDARD DEVIATION IN DEFAULT UNITS.	F7.2
6	KEYR - REAGENT LIMIT SOURCE CODE. **	A1
7	KNOWN SPIKE AMOUNT (LSA OR LS2) IN DEFAULT UNITS.	F8.2
7	SPIKE MEAN RECOVERY IN PERCENT.	F6.1
7	SPIKE STANDARD DEVIATION IN PERCENT.	F6.1
7	KEYS - SPIKE LIMIT SOURCE CODE. **	A1
8	KNOWN CONTROL STANDARD VALUE IN APPROPRIATE UNITS.	F8.2
8	CONTROL STANDARD MEAN RECOVERY IN PERCENT.	F6.1
8	CONTROL STANDARD STANDARD-DEVIATION IN PERCENT.	F6.1
8	KEYC - CONTROL STANDARD LIMIT SOURCE. **	A1/

* USE CODES:
0 - NOT SPECIFIED (NOT A QC ANALYTE)
1 - NORMAL SPIKE OR CONTROL STANDARD
2 - SURROGATE SPIKE OR CONTROL STANDARD
3 - INTERNAL STANDARD (EXCLUDES ALL QC TYPES)

** LIMIT SOURCE CODES:
D - DEFAULT LIMITS AT QC/MT CREATION TIME
A - ANALYST SUPPLIED THROUGH DMPQ
R - SUPPLIED THROUGH REPLICATE MODULE (DMPR)
M - MANUALLY ENTERED THROUGH SFC
C - COMPUTED BY SFC

number of compound records, and the compound record size. These parameters allow for easy pointer positioning in the DMPQ2 file edit procedure.

11.5 $.RD - THE REDUCED SAMPLE DATA FILE

The reduced sample data file ($.RD) is created by the DMP system and normally contains quantitation data from one measured sample. If the sample is an unknown (U), the reduced data file can also contain quantitative data on a duplicate and/or spiked unknown. For unknown (U) and non-SFC (N) samples, the reduced data file can also contain qualitative information from the qualitative identification file described above. The header and the quantitative portions of the reduced data file is shown in Tables 11.5a and 11.5b. The qualitative section of the reduced data file is identical to the $.QA file described above.

The DMPG1 program creates the reduced data file and the DMPE program can modify it. The DMPG2 program reports its contents. The DMPR programs use reduced data files to compute replicate statistics. Finally, the DMPP1 program extracts data from reduced data files for inclusion in the SFC run results file.

11.6 DMPRPTG.99 - THE MEASUREMENT AND QUALITY CONTROL REPORT

The DMPG2 program uses reduced data files and QC/MT files to generate measurement and quality control reports on various sample types. The layout of the report header and compound records for each report type are shown in Figure 11.6.

11.7 DMPRPTQ.99 - THE QUALITY CONTROL METHOD TEMPLATE CONTENTS REPORT

Since the QC/MT file is semipermanent, the DMPQ2 program generates a more elegant report on its contents. The report is an 8-1/2" x 11" paginated document containing all of the information in the file. The

TABLE 11.5A - GENERAL HEADER INFORMATION IN THE PROCESSED DATA FILE

FIELD NUMBER	PARAMETER DESCRIPTION	FORMAT
1	NUMBER OF COMPOUNDS IN LIST	I4
2	SFC SAMPLE IDENTIFIER	4A2
3	GENERIC PARMETH CODE	4A2
4	ANALYST'S INITIALS	4A2
5	EXTRACTION DATE	4A2
6	ACQUISITION DATE	4A2
8	DATA PROCESSING DATE	4A2
9	QC SAMPLE TYPE DESIGNATOR	I4/
10	INSTRUMENT IDENTIFIER	3A2
11	UNITS OF MEASURE	3A2
12	VOLUME OR MASS OF SAMPLE	F8.3
13	EXTRACT VOLUME (ML)	F8.3
14	INJECTION VOLUME (UL)	F8.3
15	QUANTITATION FACTOR	F8.3
16	COMPUTATION MODE	2A2
17	QC/METHOD TEMPLATE NAME	6A2/
18	REAGENT BLANK FILENAME	9A2
19	CONTROL STANDARD FILENAME	9A2
20	SFC RUN RESULTS FILE NAME	9A2
21	RUN RESULTS GENERATION DATE	4A2/
22	LOG SAMPLE COMMENT	32A2/
23	LOG CONDITION COMMENT	32A2/
24	USER SUPPLIED COMMENT	32A2/

TABLE 11.5B - FIXED LIST QUANTIFICATION RECORD FOR THE PROCESSED DATA FILE

FIELD NUMBER	PARAMETER DESCRIPTION	FORMAT
1	COMPOUND NUMBER	I4
2	CONCENTRATION	F12.3
3	AREA	F12.0
4	INTERNAL STANDARD REFERENCE	I5
5	RELATIVE RETENTION TIME (*1000)	I6

```
                    GC/MS COMBINED MEASUREMENT AND
                       QUALITY CONTROL REPORT
                    $$$$$$$$$$        MM/DD/YY

   ACCOUNT ID: $$$$$$$$$$                SAMPLE TYPE: $
   UNITS: $$$$$$                         QC/MT NAME: $$$$$$$
   SAMPLE VOLUME:          ####.### $$   FACTOR:             ####.###
   EXTRACTED: MM/DD/YY HH:MM             INJECTED: MM/DD/YY HH:MM
   EXTRACTION VOLUME:      ####.### ML   INJECTION VOLUME: ####.### UL
   ANALYST: $$$                          INSTRUMENT: $$$$$
   REAGENT BLANK: $$$$$$$$$$             CONTROL STANDARD: $$$$$$$$$$

              UNKNOWN AND NON-SFC SAMPLE RESULTS

                                        MEASURED       CAS
 NO  COMPOUND NAME                        CONC       NUMBER          AREA
=========================================================================
###  $$$$$$$$$$$$$$$$$$$$$$$$  ########.####  ######-##-#  ###########.
     $$$$$$$$$$$$$$$$$$$$$$$$


                                       RECOVERY/         MEASURED/     QC
 NO  SURROGATE NAME                      CAS #           EXPECTED     ASSESS
-----------------------------------------------------------------------------
###  $$$$$$$$$$$$$$$$$$$$$$$$  %REC= ####.##%   LS1= #######.###  GC-S
     $$$$$$$$$$$$$$$$$$$$$$$$  CAS ######-##-#  LS2= #######.##


                      IDENTIFICATION INFORMATION

                                          CAS
 NO  COMPOUND NAME                       NUMBER     SCAN  PURITY   FIT    S/I
-----------------------------------------------------------------------------
###  $$$$$$$$$$$$$$$$$$$$$$$$  ############ #####  ####.# ####.#   ###
     $$$$$$$$$$$$$$$$$$$$$$$$


                      GC FOR CONTROL STANDARDS

                                       RECOVERY/         MEASURED/     QC
 NO  COMPOUND NAME                       CAS #           EXPECTED     ASSESS
-----------------------------------------------------------------------------
###  $$$$$$$$$$$$$$$$$$$$$$$$  %EC= ####.##%    LCM= #######.###  GC-S
     $$$$$$$$$$$$$$$$$$$$$$$$  CAS ######-##-#  LCT= #######.##


                        GC FOR REAGENT BLANKS

                                      MEASURED/        CONTROL AND     GC
 NO  COMPOUND NAME                      CAS  #        WARNING LIMITS  ASSESS
-----------------------------------------------------------------------------
###  $$$$$$$$$$$$$$$$$$$$$$$$ LRB= #######.###  UCL= #####.##      GC-S
     $$$$$$$$$$$$$$$$$$$$$$$$ CAS   ######-##-# UWL= #####.##


                           GCFOR DUPLICATES

                                                                          GC
 NO  COMPOUND NAME              CONC.   MEASURED    MEAN/DIFFERENCE ASSESS
-----------------------------------------------------------------------------
###  $$$$$$$$$$$$$$$$$$$$$$$$ LD1= ########.###   X= ########.### $$$$$$
     $$$$$$$$$$$$$$$$$$$$$$$$ LD2= ########.###   R= ########.###


                        GC FOR NORMAL SPIKES

                                                                          GC
 NO  COMPOUND-NAME              CONC.   MEASURED    ADDED/RECOVERY   ASSESS
-----------------------------------------------------------------------------
###  $$$$$$$$$$$$$$$$$$$$$$$$ LSO= ########.###   LSA= ######.##   $$$$$$
     $$$$$$$$$$$$$$$$$$$$$$$$ LSF= ########.###   %REC= #####.#%
```

FIGURE 11.6 - OUTPUT FORMAT OF VARIOUS MEASUREMENT AND QUALITY CONTROL REPORT RECORDS

header and record layout for the report are shown in Figure 11.7.

11.8 DMPRPTR.99 - THE REPLICATE STATISTICS REPORT

There are six different types of replicate statistics reports which can be generated by the DMPR module. Since the report form is so variable, it is the only report generated by the DMP system which can exceed 74 characters in width. The header and various record layouts of the statistics report are shown in Figure 11.8.

11.9 THE SFC COMPATIBLE RUN RESULTS FILE AND THE SFC-SUPPLIED QUALITY CONTROL LIMIT FILE

Two files link the DMP system with the EPA SFC. One is the run results file created by the DMP system and sent to SFC. The other is the SFC limit file supplied to the DMP system by SFC.

The SFC compatible run results file is created by the DMPP1 program. The format of the various record types used in the report are shown in Tables 11.9a through 11.9e. Type-6 records are used to head lists of quantitative and qualitative records. Type-2 records are actual results records and type-7 records are comment records containing qualitatively identified compound names. Type-4 records are limit records which may be appended to the run results file by the DMPP2 program.

The SFC supplied limit file is the only external file read by the DMP system. Data from this file is extracted by the DMPQ3 program.

```
                        QC/METHOD TEMPLATE
                             MM/DD/YY

    NAME: PARMETH                          DEFAULT UNITS (U): UG/L
    DATE CREATED: MM/DD/YY                 LAST UPDATE:  MM/DD/YY
    CREATED BY: NAME                       CHANGED BY: NAME
    NO. OF COMPOUNDS: ###                  EDIT SUBFIELDS: (N)

COMPOUND NUMBER     1
(1) COMPOUND [NAME OF UP TO 70 CHARACTERS]
(2) CAS #: #######-##-#           (3) USE CODE: #=[DESCRIPTION]
(4) METHOD DETECTION LIMITS:      (5) DUPLICATE LIMITS /*/:
    REAGENT MDL: ####.##U         MIN [A]: ######.###  MAX [B]: ######.###
    MATRIX  MDL: ####.##U         SLP [C]: ######.###  INT [D]: ######.###

                                 EXPECTED         MEAN        STD.DEV
(6) REAGENT BLANK LIMITS:   /*/                 ####.## U    ####.## U
(7) SPIKED SAMPLE LIMITS:   /*/#####.## U        ####.# %     ####.# %
(8) CNTRL.STANDARD LIMITS:  /*/#####.## U        ####.# %     ####.# %

COMPOUND NUMBER     2
                .               .               .    .
                .               .                    .
                .               .                    .
```

FIGURE 11.7 - FORMAT OF THE QUALITY CONTROL METHOD TEMPLATE REPORT

```
                  GC/MS REPLICATE STATISTICS REPORT

   ITEM OF INTEREST: ##############  QC METHOD TEMPLATE: D:PARMETH.QC
   SAMPLE TYPE: ############          UNITS OF CONC: ######
   NUMBER OF REPLICATES:  ###         DATE COMPUTED: MM/DD/YY

                       SAMPLE IDENTIFIERS USED

##########        ##########        ##########        ##########         ##########
##########        ##########        ##########        ##########         ##########


              REPLICATE CONCENTRATION STATISTICS FOR
                   UNKNOWNS AND NON-SFC SAMPLES

                                  CAS          AVERAGE   STANDARD  % REL
  NO. COMPOUND NAME              NUMBER         VALUE   DEVIATION  STDEV
=========================================================================
 ### $$$$$$$$$$$$$$$$$$$$$$$$$ $######-##-#  ######.###  ####.### ####.#

                          CONCENTRATION, RECOVERY AND MDLR STATISTICS FOR
                                        CONTROL STANDARDS

                                  CAS          AVERAGE   STANDARD  % REL    KNOWN   MEAN  LIMIT          REAGENT
  NO. COMPOUND NAME              NUMBER         VALUE   DEVIATION  STDEV    VALUE   %REC  RANGE              MDL
===============================================================================================================
 ### $$$$$$$$$$$$$$$$$$$$$$$$$ $######-##-#  ######.###  ####.### ####.#  #####.## ####.# ####.#  #######.### $

                            RECOVERY STATISTICS FOR SPIKED SAMPLES

                                  CAS          AVERAGE   STANDARD  % REL    KNOWN   MEAN  LIMIT
  NO. COMPOUND NAME              NUMBER         VALUE   DEVIATION  STDEV    VALUE   %REC  RANGE
=================================================================================================
 ### $$$$$$$$$$$$$$$$$$$$$$$$$ $######-##-#  ######.###  ####.### ####.#  #####.## ####.# ####.#

                 MATRIX MDL STATISTICS FOR ADULTERATED UNKNOWNS

                                  CAS          AVERAGE   STANDARD  % REL       MATRIX
  NO. COMPOUND NAME              NUMBER         VALUE   DEVIATION  STDEV         MDL
=======================================================================================
 ### $$$$$$$$$$$$$$$$$$$$$$$$$ $######-##-#  ######.###  ####.### ####.#  #######.### $

         RELATIVE RETENTION TIME STATISTICS FOR ALL BUT BLANK SAMPLES

                                  CAS          AVERAGE   STANDARD  % REL  NO. OF
  NO. COMPOUND NAME              NUMBER         VALUE   DEVIATION  STDEV  ITEMS
=================================================================================
 ### $$$$$$$$$$$$$$$$$$$$$$$$$ $######-##-#  ######.###  ####.### ####.#    ##

         RESPONSE FACTOR STATISTICS FOR CALIBRATION STANDARDS

                                  CAS          AVERAGE   STANDARD  % REL
  NO. COMPOUND NAME              NUMBER         VALUE   DEVIATION  STDEV
=========================================================================
 ### $$$$$$$$$$$$$$$$$$$$$$$$$ $######-##-#  ######.###  ####.### ####.#

        RESPONSE RATIO STATISTICS FOR CALIBRATION STANDARDS

                                  CAS          AVERAGE   STANDARD  % REL
  NO. COMPOUND NAME              NUMBER         VALUE   DEVIATION  STDEV
=========================================================================
 ### $$$$$$$$$$$$$$$$$$$$$$$$$ $######-##-#  ######.###  ####.### ####.#
```

FIGURE 11.8 HEADER AND VARIOUS RECORD LAYOUTS FOR THE REPLICATE STATISTICS REPORT

TABLE 11.9A - TYPE "1" SFC RECORD

RECORD LENGTH: 80 BYTES
FILE TYPE: TRANSMISSION

RECORD POSITION	FIELD LENGTH	FIELD CONTENTS	REMARKS
1	1	RECORD TYPE	"1"
2	1	BLANK	
3-4	2	YEAR	
5	1	BLANK	
6-7	2	MONTH	
8	1	BLANK	
9-10	2	DAY	
11	1	BLANK	
12-13	2	HOUR	
14	1	BLANK	
15-16	2	MINUTES	
17	1	BLANK	
18-22	5	INSTRUMENT TYPE	(LEFT JUSTIFIED)
23	1	BLANK	
24-28	5	METHOD IDENTIFIER	
29-30	2	LABORATORY/INSTRUMENT ID	
31	1	BLANK	
32-34	3	ANALYST INITIALS	
35-80	46	NOT USED	(LEAVE BLANK)

TABLE 11.9B - TYPE "2" SFC RECORD

RECORD LENGTH: 80 BYTES
FILE TYPE: TRANSMISSION

RECORD POSITION	FIELD LENGTH	FIELD CONTENTS	REMARKS
1	1	RECORD TYPE	"2"
2	1	BLANK	
3-10	8	SAMPLE ID	(LEFT JUSTIFIED)
11	1	BLANK	
12-14	3	GC CODE	
15	1	BLANK	
16	1	PARAMETER ID CODE	"P" OR "C"
17	1	BLANK	
18-26	9	SPECIFIC PARAMETER CODE (PARMETH)	(RIGHT JUSTIFIED)
27	1	BLANK	
28-33	6	SFC RESULT	
34	1	NOTATION TYPE	"E" IF SCIENTIFIC OTHERWISE BLANK
35-37	3	EXPONENT	
38	1	BLANK	
39-46	8	UNITS OF MEASURE	
47	1	BLANK	
48-50	3	NON-NUMERIC RESULT	
51-80	30	RESERVED	

TABLE 11.9C - TYPE "4" SFC RECORD

RECORD LENGTH: 80 BYTES
FILE TYPE: TRANSMISSION

RECORD POSITION	FIELD LENGTH	FIELD CONTENTS	REMARKS
1	1	RECORD TYPE	"4"
2	1	BLANK	
3-10	8	SAMPLE ID	(LEFT JUSTIFIED)
11	1	BLANK	
12-14	3	GC CODE	
15	1	BLANK	
16	1	PARAMETER ID CODE	"P" OR "C"
17	1	BLANK	
18-26	9	SPECIFIC PARAMETER CODE (PARMETH)	(RIGHT JUSTIFIED)
27	1	BLANK	
28-33	6	NUMERIC LIMIT	
34	1	NOTATION TYPE	"E" IF SCIENTIFIC OTHERWISE BLANK
35-37	3	EXPONENT	
38	1	BLANK	
39-46	8	DATE SET	
47	1	BLANK	
48	3	INITIAL	
49-80	32	RESERVED	

TABLE 11.9D - FORMAT OF THE SFC RUN RESULTS SAMPLE HEADER RECORD

RECORD LENGTH: 80 BYTES
FILE TYPE: TRANSMISSION

RECORD POSITION	FIELD LENGTH	FIELD CONTENTS	REMARKS
1	1	RECORD TYPE	"6"
2	1	BLANK	
3-10	8	LAB SAMPLE ID	
11	1	BLANK	
12-14	3	NUMBER OF COMPOUNDS	I.E. "021"
15	1	BLANK	
16	1	"F"	
17-19	3	BLANK	
20-26	7	SPECIFIC PARMETH CODE	
27-80	54	BLANK	

TABLE 11.9E - FORMAT OF THE SFC RUN RESULTS FREE TEXT RECORD

RECORD LENGTH: 80 BYTES
FILE TYPE: TRANSMISSION

RECORD POSITION	FIELD LENGTH	FIELD CONTENTS	REMARKS
1	1	RECORD TYPE	"7"
2	1	BLANK	
3-10	8	LAB SAMPLE ID	
11	1	BLANK	
12-14	3	COMMENT SEQUENCE #	I.E. "006"
15	1	BLANK	
16	1	"P"	
17-19	3	BLANK	
20-26	7	SPECIFIC PARMETH CODE	
27	1	BLANK	
28-80	53	FREE TEXT COMMENT	

PROGRAM LISTING

```
[DMPL.LD     8/82     LOG FILE MODIFICATION ROUTINE]
GL
L/BF CMAIN+MFLIB+DMPL+MEMLM+
TIMER+ISHFT+NCHAR+LENG+MOVBT+NULST+
PADST+INPTS+NPOS+MAPST+VALUE+GOBBL+
INPTF+INPTI
M/TU
M MAP
S/F 2:DMPL.MS.CJNK2:.END
E
```

```
[DMPN.LD   8/82   DMP QUAN.99 RENAME MODULE]
GL
L/BF CMAIN+MFLIB+DMPN+ISHFT+
FCOPY+LENG+MOVBT+MEMLM
M/TU
M MAP
S/F 2:DMPN.MS.CDESC:.END
E
```

```
[DMPC.LD     8/82   DMP MASTER COMMAND ROUTER]
GL
L/BF CMAIN+MFLIB+DMPC+MEMLM+
TIMER+ISHFT+NCHAR+LENG+MOVBT+NULST+
PADST+INPTS+NPOS+MAPST
M/TU
M MAP
S/F 2:DMPC.MS.CJNK2:.END
E
```

```
[DMPG1.LD     9/1/81     TEST OF LOG FILE READ ROUTINE]
GL
L/BF CMAIN+MFLIB+DMPG1+FSRT2+FMOPN+FMRD+FMCLS+MEMLM+
TIMER+ISHFT+NCHAR+LENG+MOVBT+NULST+
PADST+INPTS+NPOS+MAPST+VALUE+GOBBL
M/TU
M MAP
S/F 2:DMPG1.MS.CJNK2:.END
E
```

```
[DMPG2.LD     5/6/82   REPORT GENERATOR]
GL
L/BF CMAIN+MFLIB+DMPG2+FSRT2+FMOPN+FMRD+
FMCLS+MEMLM+FCOPY+OPLPT+TIMER+ISHFT+
NCHAR+LENG+MOVBT+NULST+PADST+INPTS+
NPOS+MAPST+VALUE+GOBBL+MOVST+GCTST
M/TU
M MAP
S/F 2:DMPG2.MS.CJNK2:.END
E
```

```
[DMPG1.LD   2/82   GC/MT EDITOR: GENERATE SKELETON WITH DEFAULTS]
GL
L/BF CMAIN+MFLIB+DMPG1+FSRT2+TIMER+ISHFT+MEMLM+
LENG+NCHAR+MOVBT+INPTF+NULST+
NPOS+PADST+INPTS+MOVST+MAPST+GOBBL
M/TU
M MAP
S/F 2:DMPG1.MS,CJNK2:.END
E


[DMPG2.LD   2/82   GC/MT EDITOR: COMMAND LINE AND REPORTER]
GL
L/BF CMAIN+MFLIB+DMPG2+FSRT2+TIMER+ISHFT+MEMLM+
OPLPT+FCOPY+MEMLM+GOSUB+
LENG+NCHAR+MOVBT+INPTF+NULST+INPTI+
NPOS+PADST+INPTS+MOVST+GOBBL+NCAS
M/TU
M MAP
S/F 2:DMPG2.MS,CJNK2:.END
E


[DMPR1.LD   2/82   REPLICATE ANALYSIS: GET NAMES]
GL
L/BF CMAIN+MFLIB+DMPR1+FCOPY+FMCHK+
FMOPN+FMWRT+FMCLS+FMRD+OPLPT+
MEMLM+FSRT2+TIMER+ISHFT+
NCHAR+LENG+MOVBT+NULST+
PADST+INPTS+NPOS+MAPST+GOBBL
M/TU
M MAP
S/F 2:DMPR1.MS,CJNK2:.END
E


[DMPR2.LD   2/82   REPLICATE ANALYSIS: COMPUTE STATISTICS]
GL
L/BF CMAIN+MFLIB+DMPR2+FCOPY+FMCHK+
FMOPN+FMWRT+FMCLS+FMRD+OPLPT+
MEMLM+FSRT2+TIMER+ISHFT+
NCHAR+LENG+MOVBT+NULST+
PADST+INPTS+NPOS+MAPST+GOBBL
M/TU
M MAP
S/F 2:DMPR2.MS,CJNK2:.END
E


[DMPR3.LD   2/82   REPLICATE ANALYSIS: GENERATE REPORT]
GL
L/BF CMAIN+MFLIB+DMPR3+FCOPY+FMCHK+
FMOPN+FMWRT+FMCLS+FMRD+OPLPT+
MEMLM+FSRT2+TIMER+ISHFT+
NCHAR+LENG+MOVBT+NULST+
PADST+INPTS+NPOS+MAPST+GOBBL
M/TU
M MAP
S/F 2:DMPR3.MS,CJNK2:.END
E
```

```
[DMPP1.LD     6/30/82   RUN RESULTS FILE PACKAGER - RESULTS GENERATOR]
GL
L/BF CMAIN+MFLIB+DMPP1+FSRT2+FMOPN+FMRD+
FMCLS+MEMLM+FCOPY+OPLPT+TIMER+ISHFT+
NCHAR+LENG+MOVBT+NULST+PADST+INPTS+
NPOS+MAPST+VALUE+GOBBL+MOVST+GCTST
M/TU
M MAP
S/F 2:DMPP1.MS,CJNK2:.END
E
```

```
[DMPP2.LD     6/30/82   RUN RESULTS FILE PACKAGER - RESULTS ROUTER]
GL
L/BF CMAIN+MFLIB+DMPP2+FSRT2+FMOPN+FMRD+
FMCLS+MEMLM+FCOPY+OPLPT+TIMER+ISHFT+
NCHAR+LENG+MOVBT+NULST+PADST+INPTS+
NPOS+MAPST+VALUE+GOBBL+MOVST+GCTST
M/TU
M MAP
S/F 2:DMPP2.MS,CJNK2:.END
E
```

```
C       DMPL.FR   ASSISTS USER IN MODIFYING LOG FILE
C       8/11/82   D.RYAN UCCC     LAST REVISION: 12/08/82

        PARAMETER TTO=10, ICH=2

        COMMON /CJNK1/
     1  ITMP(20), ITXT(50), IANS(5),
     1  IYR, IMO, IDAY, IHR, IMIN, ISEC, IX, IFOUND, I, ICRDF, IPOS, ICODE,
     1  LN, LNN, IP1, VOL, VEX, VINJ, ISOURCE, IER, IE, XTMP

        COMMON /CJNK2/
     1  MCODE(4), MATCH(8), IEXTL(2), LOOK2(2), IVS(2), IEXDT(4), IPMTH(4),
     1  IPVTH(4), ICTST(5), IRGBK(5), ICMNT(32)

        COMMON /CJNK3/
     1  ITIME(9), IFLNM(8), IFLFN(10)

        COMMON /CMAIN/
     1  NPROG, IDFSW, IDFSW1, ITEMP(15),
     1  IDFILE(12,6), ISFILE(11,4),
     1  INSTD(3), IREFTABLE(11),
     1  IADD1, IADD2, ISUB1, ISUB2, ISUB3, ISUB4,
     1  JADD1, LIBRNAME, LIBRNUMB, IBUF,
     1  ISCN(45), IACG(100), IDAT(45)

        DATA MCODE/"UDFXCRNS"/
        DATA MATCH/"VS<75> MD<75> MW<75> AS<75>"/
        DATA IVS/"VS", "<75><0>"/
        DATA IEXDT/"MM<57>DD<57>YY"/
        DATA IPMTH/4*20040K/
        DATA IPVTH/4*20040K/
        DATA ICTST/5*20040K/
        DATA IRGBK/5*20040K/
        DATA ICMNT/32*20040K/
        DATA IEXTL/".LG<0>"/
        DATA LOOK2/"$$$<0>"/

        CALL GETTI(IYR, IMO, IDAY, IHR, IMIN, ISEC, $50, IX, IX, IX, IX, IX, IX)
        ITIME(1)=(IMO/10+48)*256+MOD(IMO,10)+48
        ITIME(2)=47*256+IDAY/10+48
        ITIME(3)=(MOD(IDAY,10)+48)*256+47
        ITIME(4)=(MOD(IYR/10,10)+48)*256+MOD(IYR,10)+48
        ITIME(5)=20040K
        ITIME(6)=(IHR/10+48)*256+MOD(IHR,10)+48
        ITIME(7)=58*256+IMIN/10+48
        ITIME(8)=(MOD(IMIN,10)+48)*256+58
        ITIME(9)=(ISEC/10+48)*256+MOD(ISEC,10)+48
        GOTO 100

50      WRITE(TTO) " <7>PLEASE SET TIME"
        CALL LOADIT("EXEC")

100     CALL NULST(IFLNM,8,1)
        DO 120 I=4,9
            IFLNM(I-3)=ISFILE(I,1)
120     CONTINUE

        WRITE(TTO,8000)

        IFOUND=0
        CALL INPTS(ITXT, IFLNM, 8, $8100, $2000)
        CALL NULST(IFLFN,1,10)
        DO 200 I=1,6
            IFLFN(I)=IFLNM(I)
200     CONTINUE

        LN=LENG(IFLNM,8)
        LNN=LN+1
        ICRDF=NCHAR(IFLNM,8,LN)
        IPOS=NPOS(ICRDF,1,MCODE,4,1)
        ICODE=IPOS-1
        IF(ICODE.LT.0) ICODE=6
        CALL MAPST(IEXTL,1,3,2,IFLFN,10,LNN,$9999)

        CALL OPENR(IFLFN, ICH, IER, $300)
        GOTO 1000
```

```
300         CALL RESET
            WRITE(TTO) " <7>LOG FILE ACCESS ERROR."
            GOTO 100

1000        READ(ICH, 8300, END=1240, ERR=1240) (ITXT(I), I=1, 40)
            IP1=NPOS(LOOK2, 2, ITXT, 5, 1)
            IF(IP1. LE. 0) GOTO 1000

C           *** FOUND
            IFOUND=1
            READ(ICH, 8400, END=1240, ERR=1240) IVS, VOL, VEX, IEXDT, VINJ
            READ(ICH, 8420, END=1240, ERR=1240) IPMTH, ICTST, IRGBK, IPVTH
            READ(ICH, 8440, END=1240, ERR=1240) ICMNT
            ISOURCE=(NPOS(IVS, 2, MATCH, 8, 1)+3)/4
            IF(ISOURCE. EQ. 0) GOTO 1240

            CALL POSITION(ICH, 1, -1, -1, IER, $1200)
1200        CALL CLOSE(ICH, $1220)
1220        WRITE(TTO) " LOG FILE HAS PARAMETERS.  THEY WILL BE THE DEFAULTS"
            GOTO 1400

1240        ISOURCE=1
            IVS(1)="VS"
            VOL=1. 0
            VEX=1. 0
            VINJ=1. 0

1400        CALL INPTI(I, ISOURCE, $8500, $2000)
            IF(ISOURCE. GE. 1. AND. ISOURCE. LE. 4) GOTO 1420
            WRITE(TTO) " <7>"
            GOTO 1400

1420        IF(ICODE. EQ. 7) GOTO 1660
            XTMP=VOL
            GOTO(1460, 1500, 1540, 1580) ISOURCE
            GOTO 9999

1460        IVS(1)="VS"
            CALL INPTF(VOL, XTMP, $8520, $2000)

1480        IF(XTMP. GT. 0. 0005) GOTO 1620
            WRITE(TTO) " <7>"
            GOTO 1420

1500        IVS(1)="MD"
            CALL INPTF(VOL, XTMP, $8540, $2000)
            GOTO 1480

1540        IVS(1)="MW"
            CALL INPTF(VOL, XTMP, $8560, $2000)
            GOTO 1480

1580        IVS(1)="AS"
            CALL INPTF(VOL, XTMP, $8580, $2000)
            GOTO 1480

1620        XTMP=VEX
            CALL INPTF(VEX, XTMP, $8600, $2000)
            IF(VEX. GT. 0. 0005) GOTO 1640
            WRITE(TTO) " <7>"
            GOTO 1620

1640        CALL INPTS(ITMP, IEXDT, 4, $8620, $2000)
            CALL PADST(IEXDT, 4)

1660        XTMP=VINJ
            CALL INPTF(VINJ, XTMP, $8640, $2000)
            IF(VINJ. GT. 0. 0005) GOTO 1680
            WRITE(TTO) " <7>"
            GOTO 1660
```

```
1680    CALL INPTS(ITMP,IPMTH,4,$8660,$2000)
        CALL PADST(IPMTH,4)
        IF(ICODE.NE.0) GOTO 1700

        CALL INPTS(ITMP,ICTST,5,$8680,$2000)
        CALL PADST(ICTST,5)

        CALL INPTS(ITMP,IRGBK,5,$8700,$2000)
        CALL PADST(IRGBK,5)

        CALL INPTS(ITMP,IPVTH,4,$8720,$2000)
        CALL PADST(IPVTH,4)

1700    CALL INPTS(ITXT,ICMNT,32,$8740,$2000)
        CALL PADST(ICMNT,32)

        VOL=VOL+0.0005
        VEX=VEX+0.0005
        VINJ=VINJ+0.0005
        WRITE(TTO,8750)
        WRITE(TTO,8760) IVS,VOL,VEX,IEXDT,VINJ
        WRITE(TTO,8780) IPMTH,ICTST,IRGBK,IPVTH
        WRITE(TTO,8800) ICMNT

1720    IANS(1)="Y<0>"
        CALL INPTS(ITMP,IANS,1,$8820,$2000)
        IF(IANS(1).EQ."N<0>") GOTO 1400
        IF(IANS(1).EQ."Y<0>") GOTO 1760
        WRITE(TTO) " <7>"
        GOTO 1720

1760    CALL OPEN(IFLFN,ICH,IER,$9999)
        IF(IFOUND.EQ.0) GOTO 1800

1780    READ(ICH,8300,END=9999,ERR=9999) (ITXT(I),I=1,40)
        IP1=NPOS(LOOK2,2,ITXT,5,1)
        IF(IP1.LE.0) GOTO 1780
        GOTO 1840

1800    CALL POSITION(ICH,1,-1,-1,IER,$1820)
1820    WRITE(ICH,8750)
        WRITE(ICH,8840) ITIME
        WRITE(ICH,8860)

1840    WRITE(ICH,8760) IVS,VOL,VEX,IEXDT,VINJ
        WRITE(ICH,8780) IPMTH,ICTST,IRGBK,IPVTH
        WRITE(ICH,8800) ICMNT

        CALL POSITION(ICH,1,-1,-1,IER,$1860)
1860    CALL CLOSE(ICH,IE)
        CALL LOADIT("EXEC")

2000    WRITE(TTO) " <7> TASK ABORTED"
        CALL LOADIT("EXEC")
```

```
8000    FORMAT("1","LOG FILE MODIFICATION ROUTINE (DMPL)")
8100    FORMAT("  SAMPLE NAME: [",8A2,"] > ",Z)
8300    FORMAT(40A2)
8400    FORMAT(2A2,F10.3,6X,F10.3,6X,4A2,6X,F10.3,1X)
8420    FORMAT(4X,4A2,6X,5A2,6X,5A2,6X,4A2,1X)
8440    FORMAT(4X,32A2)
8500    FORMAT(" ",/,"  SAMPLE SOURCE CODE:",/,
     1            "    1 - WATER",/,
     1            "    2 - DRY MATTER (SOIL)",/,
     1            "    3 - WET MATTER (SEDIMENT)",/,
     1            "    4 - AIR",/,
     1            "  OPTION: [",I4,"] > ",Z)
8520    FORMAT("  SAMPLE VOLUME:  [",F10.3,"] LITERS         > ",Z)
8540    FORMAT("  SAMPLE VOLUME:  [",F10.3,"] DRY KILOGRAMS > ",Z)
8560    FORMAT("  SAMPLE VOLUME:  [",F10.3,"] WET KILOGRAMS > ",Z)
8580    FORMAT("  SAMPLE VOLUME:  [",F10.3,"] CUBIC METERS  > ",Z)
8600    FORMAT("  EXTRACT VOLUME: [",F10.3,"] MILILITERS    > ",Z)
8620    FORMAT("  EXTRACTION DATE: [",4A2,"]                > ",Z)
8640    FORMAT("  INJECTN VOLUME: [",F10.3,"] MICROLITERS   > ",Z)
8660    FORMAT("  FIXED LIST PARMETH: [",4A2,"]             > ",Z)
8680    FORMAT("  CONTROL STANDARD REFERENCE: [",5A2,"]     > ",Z)
8700    FORMAT("  REAGENT BLANK REFERENCE:    [",5A2,"]     > ",Z)
8720    FORMAT("  VARIABLE LIST PARMETH: [",4A2,"]          > ",Z)
8740    FORMAT("  ADDITIONAL COMMENTS: ",/,
     1            "     [",32A2,"]",/,
     1            "  COMMENT > ",Z)
8750    FORMAT(" ")
8760    FORMAT(" ",2A2,F10.3,"  VE= ",F10.3,"  XD= ",4A2,"  VI= ",F10.3)
8780    FORMAT(" PM= ",4A2,"  CS= ",5A2,"  RB= ",5A2,"  PV= ",4A2)
8800    FORMAT(" CM= ",32A2)
8820    FORMAT("  ACCEPTABLE LOG FILE PARAMETERS [",A2,"] > ",Z)
8840    FORMAT(" ",9A2)
8860    FORMAT(" $$$")

9999    CALL RESET
        WRITE(TTO) " <7>FATAL ERROR EXIT"
        CALL LOADIT("EXEC")

        END
```

```
C       DMPN.FR          COPIES QUAN.99 INTO $.QN
C       4/14/82        LAST REVISION: 09/23/82

        PARAMETER TTO=10

        COMMON /CDESC/ IDESC(11),JDESC(11),IQNFN(10)

        COMMON /CMAIN/
     1  NPROG,IDFW,IDFW1,ITEMP(15),
     1  IDFILE(12,6),ISFILE(11,4),
     1  INSTD(3),IREFTABLE(11),
     1  IADD1,IADD2,ISUB1,ISUB2,ISUB3,ISUB4,
     1  JADD1,LIBRNAME,LIBRNUMB,IBUF,
     1  ISCN(45),IACQ(100),IDAT(45)

        DATA JDESC /"1<0>",0,1,"QUAN",0,0,0,0,"99",0/

        DO 100 I=1,11
            IDESC(I)=ISFILE(I,1)
100     CONTINUE
        IDESC(10)="QN"

        DO 200 I=4,9
            IQNFN(I-2)=IDESC(I)
200     CONTINUE
        IF(IDESC(1).EQ.0) IDESC(1)="D<0>"
        CALL MOVBT(IDESC(1),IQNFN,10,1,$500)
        CALL MOVBT(":<0>",IQNFN,10,2,$500)
        LQN1=LENG(IQNFN,10)+1
        IF(LQN1.LE.3) GOTO 500
        LQN2=LQN1+1
        LQN3=LQN2+1
        CALL MOVBT(".<0>",IQNFN,10,LQN1,$500)
        CALL MOVBT("Q<0>",IQNFN,10,LQN2,$500)
        CALL MOVBT("N<0>",IQNFN,10,LQN3,$500)
        GOTO 600

500     WRITE(TTO) " <7>MOVBYTE ERROR ...  PUNT"
        GOTO 6000

600     KCHAN=3
        ICHAN=4
        CALL OPENR("1:QUAN.99",ICHAN,IER,$3000)
        CALL OPENW(IQNFN,KCHAN,IER,$2000)
        CALL FCOPY(ICHAN,KCHAN,$1000,IE,IER)
1000    CALL CLOSE(ICHAN,IEE)
        CALL CLOSE(KCHAN,IEE)
        WRITE(TTO,1020) (IQNFN(I),I=1,10),
1020    FORMAT(" QUAN.99 CHANGED TO:  ",10A2)
        CALL DELETE("1:QUAN.99",IER,$3000)
        GOTO 6000

2000    CALL RESET
        WRITE(TTO,2020) IER
2020    FORMAT(" <7>$.QN ERROR:  ",OI6)
        GOTO 6000

3000    CALL RESET
        WRITE(TTO,3020) IER
3020    FORMAT(" <7>QUAN.99 ERROR:  ",OI6)

6000    DO 6040 I=1,1000
            X=FLOAT(I)
            X=SIN(X)/X
6040    CONTINUE

        CALL LOADIT("EXEC")

        END
```

```
C       DMPC.FR    DMP MASTER COMMAND ROUTER
C       8/82    D.RYAN UCCC    LAST REVISED 9/9/83 FOR QC/MT INIT FIX

        PARAMETER TTO=10, ICH=2

        COMMON /CJNK1/
     1  ITMP(20), IANS(5),
     1  IYR, IMO, IDAY, IHR, IMIN, ISEC, IX, IG, IE, IR, IP,
     1  IQ, IU, IEE, ILEN, ICOMND, ICMDNM

        COMMON /CJNK2/
     1  IEXTC(2), ICMDST(4)

        COMMON /CJNK3/
     1  IQCFN(10)

        COMMON /CJNK4/
     1  IQCNM(4), IPRDT(4)

        COMMON /CMAIN/
     1  NPROG, IDFSW, IDFSW1, ITEMP(15),
     1  IDFILE(12,6), ISFILE(11,4),
     1  INSTD(3), IREFTABLE(11),
     1  IADD1, IADD2, ISUB1, ISUB2, ISUB3, ISUB4,
     1  JADD1, LIBRNAME, LIBRNUMB, IBUF,
     1  ISCN(45), IACQ(100), IDAT(45)

        DATA ICMDST/"GERPQUXH"/
        DATA IEXTC/".QC<0>"/

        CALL GETTI(IYR, IMO, IDAY, IHR, IMIN, ISEC, *50, IX, IX, IX, IX, IX, IX)
        ITMP(1)=(IMO/10+48)*256+MOD(IMO,10)+48
        ITMP(2)=47*256+IDAY/10+48
        ITMP(3)=(MOD(IDAY,10)+48)*256+47
        ITMP(4)=(MOD(IYR/10,10)+48)*256+MOD(IYR,10)+48
        GOTO 100

50      WRITE(TTO) " <7>PLEASE SET TIME"
        CALL LOADIT("EXEC")

100     WRITE(TTO,8000)
        CALL NULST(IQCFN,1,10)
        IG=1
        IE=0
        IR=1
        IP=1
        IG=1
        IU=0
        DO 140 I=1,4
             IF(IPRDT(I).NE.ITMP(I)) GOTO 150
140     CONTINUE
        GOTO 180

150     DO 160 I=1,4
             IPRDT(I)=ITMP(I)
160     CONTINUE
        CALL NULST(ITMP,1,20)
        IQCNM(1)="XX"
        IQCNM(2)="XY"
        IQCNM(3)="ZM"
        IQCNM(4)="N<0>"
        CALL INPTS(ITMP, IQCNM, 4, *8040, *7000)

        IEE=77K
        ILEN=LENG(IQCNM,4)
        IF(ILEN.EQ.7) GOTO 180
        WRITE(TTO,8020)
        GOTO 150
```

```
180     DO 200 I=1,4
            IQCFN(I)=IQCNM(I)
200     CONTINUE
        CALL MAPST(IEXTC,1,3,2,IQCFN,10,8,$9999)

        CALL OPENR(IQCFN,ICH,IE   $260)
        CALL CLOSE(ICH,IE)
        GOTO 300


260     CALL RESET
        WRITE(TTO,8080) IQCNM

300     ICOMND="H<0>"
        CALL INPTS(ITMP,ICOMND,1,$8100,$7000)
        ICMDNM=NPOS(ICOMND,1,ICMDST,4,1)

        IF(ICMDNM.LE.0) GOTO 500

C               G    E    R    P    Q    U    X   H
        GOTO (1000,2000,3000,4000,5000,6000,7000,500) ICMDNM

500     WRITE(TTO,8200)
        GOTO 300

1000    IF(IG.NE.1) GOTO 1900
        CALL LOADIT("DMPG1")

1900    WRITE(TTO,8220)
        GOTO 300

2000    IF(IE.NE.1) GOTO 1900
        CALL LOADIT("DMPE")

3000    IF(IR.NE.1) GOTO 1900
        CALL LOADIT("DMPR1")

4000    IF(IP.NE.1) GOTO 1900
        CALL LOADIT("DMPP1")

5000    IF(IQ.NE.1) GOTO 1900
        CALL LOADIT("DMPQ1")

6000    IF(IU.NE.1) GOTO 1900
        CALL LOADIT("DMPU")

7000    CALL LOADIT("EXEC")

8000    FORMAT(
     1  "1**************************************************",/,
     1  " *  DATA MANAGEMENT PROCESS COMMAND LEVEL (DMPC)  *",/,
     1  " **************************************************",/)
8020    FORMAT(" <7>QC<57>MT NAME MUST BE SEVEN CHARACTERS LONG",//)
8040    FORMAT("  QC<57>METHOD TEMPLATE NAME: [",4A2,"]  > ",Z)
8080    FORMAT(" <7>YOU DO NOT CURRENTLY HAVE QC<57>MT: ",4A2,/,
     1          " IF YOU HAVE NEVER CREATED A QC<57>MT OR WISH",/,
     1          " TO DO SO WITH THE ABOVE NAME, PLEASE SELECT",/,
     1          " MENU OPTION Q BELOW",//)
8100    FORMAT("  DMP COMMAND (GERPQUXH): [",A2,"]  > ",Z)
8200    FORMAT(/," DMP MASTER COMMAND SUMMARY:",/,
     1          "   G - GATHER SAMPLE DATA AND GENERATE REPORT",/,
     1          "   E - EDIT REDUCED DATA FILE",/,
     1          "   R - COMPUTE REPLICATE STATISTICS",/,
     1          "   P - PACKAGE AND SEND SFC RUN RESULTS",/,
     1          "   Q - CREATE AND<57>OR EDIT QC<57>MT",/,
     1          "   U - UTILITY OPTIONS",/,
     1          "   X - EXIT TO MSDS",/,
     1          "   H - HELP PRINTS THIS SUMMARY")
8220    FORMAT(" <7>COMMAND OPTION IS NOT IMPLEMENTED")
```

```
9999    CALL RESET
        WRITE(TTO) " <7>FATAL ERROR EXIT"
        CALL LOADIT("EXEC")

        END
```

```
C       DMPG1.FR     DATA MANAGEMENT REPORT CONSOLIDATOR.
C       D.RYAN 9/81     LAST REVISION:  09/23/82

        PARAMETER TTO=10,LPT=2,DSK=1
        PARAMETER MAXN=120,MAXSM=25

        COMMON /CJNK1/
     1  IRPDT(4),KVOL(2),IRRNM(9),IRRDT(4),
     1  INST(3),ISMPLE(32),IANLST(4),ICOND(32),IACCNT(4),
     1  ITMP(20),ITXT(50),IACDT(4),IACDT2(4),KUNITS(3),
     1  IRUN(10),NAME(35),METH(2),IPMTH(4),IPVTH(4),
     1  IEXDT(4),IEXDT2(4),ICKST(5),IRGBK(5),ICMNT(32),
     1  NCOMP(2),ICOMP(64),ICAS(6),IDATA(MAXN,2),SDATA(MAXN,2),IVS(2),
     1  ICRU,ICRGS,ICREG,NIND,ISAMP,IYR,IMO,IDAY,IHR,IMIN,ISEC,IX,INM,
     1  IC,IEE,IER,NSAMP,LTYP,LOC,ICODE,ICRDF,NPIND,ILEN,IANS,LNN,LN,
     1  IPOS,IPR,ILINE,IP1,IP2,IP3,LP,LP1,LTX,I,I1,I2,I3,IND,IREF,KO,KP1,
     1  RRT,VOL,VEX,VINJ,VX,X,FACTOR,F,CONC,D1,D2,D3,D4,X1,X2,X3,X4,
     1  AR,AM,ICH,LCH,MCH,ISOURCE

        COMMON /CJNK2/
     1  MUNITS(20),MATCH(6),MCODE(4),LOOK1(4),LOOK2(2),LOOK3(5),LOOK4(3),
     1  LOOK5(4),LOOK6(4),LOOK7(4),IEXTC(2),IEXTN(2),IEXTR(2),IEXTG(2),
     1  IEXTA(2)

        COMMON /CJNK3/
     1  IGCDSC(11),IGCFN(10),IFLDSC(11),IFLNM(8),IFLFN(10),
     1  INLDSC(11),INLNM(8),IS(11,MAXSM),LSTMD

        COMMON /CJNK4/
     1  IGCNM(4),IPRDT(4)

        COMMON /CMAIN/
     1  NPROG,IDFSW,IDFSW1,ITEMP(15),
     1  IDFILE(12,6),ISFILE(11,4),
     1  INSTD(3),IREFTABLE(11),
     1  IADD1,IADD2,ISUB1,ISUB2,ISUB3,ISUB4,
     1  JADD1,LIBRNAME,LIBRNUMB,IBUF,
     1  ISCN(45),IACG(100),IDAT(45)

        DATA MUNITS/"UG","<57>L","  ","UG","<57>K","GD","UG","<57>K",
     1  "GW","UG","<57>C","M ",0,0,0,0,0,0,0,0/
        DATA MATCH/"VS","<75>M","D<75>","MW","<75>A","S<75>"/
        DATA MCODE/"UD","FX","CR","NS"/
        DATA LOOK1/"AC","GU","IR","E<0>"/
        DATA LOOK2/"$$$<0>"/
        DATA LOOK3/"NO","T ","FO","UN","D<0>"/
        DATA LOOK4/"AM","OU","NT"/
        DATA LOOK5/"NO","  ","NA","ME"/
        DATA LOOK6/"NO","  ","M<57>","E "/
        DATA LOOK7/"NO","  ","RE","T("/
        DATA IEXTC/".G","C<0>"/
        DATA IEXTN/".G","N<0>"/
        DATA IEXTG/".L","G<0>"/
        DATA IEXTR/".R","D<0>"/
        DATA IEXTA/".G","A<0>"/

        D1=0.5
        D2=0.05
        D3=0.005
        D4=0.0005
        ICRU="U<0>"
        ICRGS="?<0>"
        ICREG="=<0>"
        ICRDF="N<0>"
        KCHAN=1
        NIND=0
        KO=0
        ISAMP=0
        NSAMP=0
        ICH=3
        LCH=4
        MCH=5
        CALL NULST(ITMP,1,20)
        CALL NULST(ITXT,1,50)
        CALL NULST(IGCDSC,1,11)
        CALL NULST(IFLDSC,1,11)
        CALL NULST(INLDSC,1,11)
        NCOMP(1)="IN"
        NCOMP(2)="TS"
        CALL GETTI(IYR, IMO, IDAY, IHR, IMIN, ISEC, $50, IX, IX, IX, IX, IX, IX)
        IPRDT(1)=(IMO/10+48)*256+MOD(IMO, 10)+48
        IPRDT(2)=47*256+IDAY/10+48
        IPRDT(3)=(MOD(IDAY, 10)+48)*256+47
        IPRDT(4)=(MOD(IYR/10, 10)+48)*256+MOD(IYR, 10)+48
        GOTO 140

50      WRITE(TTO) " <7>SET TIME AT MSDS."
        GOTO 9020

C       GET CURRENT * NAME
C
140     DO 150 I=1, 11
            IFLDSC(I)=ISFILE(I, 1)
150     CONTINUE
        IFLDSC(1)="D<0>"
        IFLDSC(3)=1
        IFLDSC(10)="GN"

C       SALUTATION
C
        WRITE(TTO, 8000)

C       GET GC/METHOD TEMPLATE NAME
C
200     IEE=777K
        CALL INPTS(ITMP, IGCNM, 4, $8010, $9020)
        ILEN=LENG(IGCNM, 4)
        IF(ILEN.NE.7) GOTO 1140
        DO 240 I=1, 4
            IGCFN(I)=IGCNM(I)
240     CONTINUE
C       APPEND .GC TO FILENAME
        CALL MAPST(IEXTC, 1, 3, 2, IGCFN, 10, 8, $1140)
        IGCDSC(1)="D<0>"
        IGCDSC(3)=1
        IGCDSC(10)="GC"
        DO 1130 I=4, 7
            IGCDSC(I)=IGCNM(I-3)
1130    CONTINUE
        CALL FSLKU(IGCDSC, $1140, $9000, IEE)
        GOTO 1160

C       GC/MT NO ACCEPTABLE
1140    WRITE(TTO, 8020) IEE
        GOTO 200

C       GET MODE: MANUAL OR NAMELIST
1150    WRITE(TTO) " <7>"
1160    IANS="M<0>"
        CALL INPTS(LSTMD, IANS, 1, $8060, $9020)
        IF(IANS.NE."L<0>".AND.IANS.NE."M<0>") GOTO 1150

        IF(LSTMD.EQ."M<0>") GOTO 1550

C       GET NAMELIST AND READ
C
1200    CALL NULST(INLNM, 1, 8)
        DO 1210 I=1, 6
            INLNM(I)=IDFILE(I+3, 1)
1210    CONTINUE
        CALL INPTS(ITMP, INLNM, 8, $8080, $9020)
        DO 1230 I=4, 9
            INLDSC(I)=INLNM(I-3)
1230    CONTINUE
        INLDSC(1)="D<0>"
        INLDSC(3)=1
        INLDSC(10)="NL"
        CALL FSLKU(INLDSC, $1240, $9000, IEE)
        GOTO 1260

C       NAMELIST DOES NOT COMPUTE
1240    WRITE(TTO, 8100) IEE
        GOTO 1200
```

```
1260    IC=0
        CALL FMOPS(INLDSC, KCHAN, $1240, IER)
        CALL FMRD(IS, KCHAN, 11*MAXSM, $1270, IEE)
        IEE=0
1270    CALL FMCLS(KCHAN, $1280)
1280    IF(IEE. GT. 0) GOTO 1240
        IC=(11*MAXSM+IEE)/11
        NSAMP=IC

C       GET NEXT NAME FROM LIST
C
1500    CALL NULST(IFLNM, 1, 8)
        ISAMP=ISAMP+1
        IF(ISAMP. GT. NSAMP) GOTO 7000
        DO 1520 I=4, 9
            I1=I-3
            IFLNM(I1)=IS(I, ISAMP)
1520    CONTINUE
        GOTO 1590

C       GET NEXT MANUALLY ENTERED NAME
C
1550    DO 1560 I=1, 6
            IFLNM(I)=IFLDSC(I+3)
1560    CONTINUE
        CALL INPTS(ITXT, IFLNM, 8, $8120, $9020)

C       PREPARE FILENAME FOR PROCESSING
C
1590    CALL NULST(IFLFN, 1, 10)
        DO 1600 I=1, 6
            I1=I+3
            IFLDSC(I1)=IFLNM(I)
            IFLFN(I)=IFLNM(I)
1600    CONTINUE

C       FIND KEY CHARACTER AND SET SAMPLE TYPE CODE
C
        LN=LENG(IFLNM, 8)
        LNN=LN+1
        LOC=LN+6
        ICRDF=NCHAR(IFLNM, 8, LN)
        IPOS=NPOS(ICRDF, 1, MCODE, 4, 1)
        ICODE=IPOS-1
        IF(ICODE. LT. 0) ICODE=6
        IF((ICODE. NE. 1). AND. (ICODE. NE. 2)) GOTO 1620
        WRITE(TTO, 8140)
        IER=0
        GOTO 1610

C       CHECK OUT GN AND LG FILES
C
1620    IFLDSC(1)="D<0>"
        IFLDSC(2)=0
        IFLDSC(3)=1
        IFLDSC(10)="GN"
        WRITE(TTO, 8150) (IFLDSC(I), I=4, 9)
        CALL FSLKU(IFLDSC, $1610, $1610, IER)
        GOTO 1630

C       HANDLE LG/GN ACCESS ERRORS HERE
C
1610    WRITE(TTO, 8160) IFLDSC(1), (IFLDSC(I), I=4, 9), IFLDSC(10), IER
        IF(LSTMD. EQ. "L<0>") GOTO 1500
        GOTO 1550

1630    IFLDSC(10)="LG"
        CALL FSLKU(IFLDSC, $1610, $1610, IER)
C       ATTACH LG TO FILE DESCRIPTOR
        CALL MAPST(IEXTG, 1, 3, 2, IFLFN, 10, LNN, $1610)
```

```
C       READ LOG FILE
C
        CALL OPENR(IFLFN, ICH, IER, $1610)
        CALL NULST(KUNITS, 1, 3)
        CALL NULST(IACDT, 1, 4)
        CALL NULST(ISMPLE, 1, 32)
        CALL NULST(ICOND, 1, 32)
        CALL NULST(INST, 1, 3)
        CALL NULST(IANLST, 1, 4)
        CALL NULST(IACCNT, 1, 4)
        KO=1
        VOL=0.
        VEX=0.
        VINJ=0.
        VX=0.
        IPR=OK
        ILINE=OK
        IP1=0

C       LOOK FOR ACGUIRE
C
1700    READ(ICH, 8210, END=1779, ERR=1779) (ITXT(I), I=1, 40)
        IP1=NPOS(LOOK1, 4, ITXT, 10, 1)
        IF(IP1. LE. 0) GOTO 1700

C       READ LOG FILE HEADER
C
        READ(ICH, 8200, END=1779, ERR=1779) IACDT
        READ(ICH, 8220, END=1779, ERR=1779) ISMPLE
        READ(ICH, 8220, END=1779, ERR=1779) ICOND
        INST(1)="FI"
        INST(2)="NN"
        INST(3)="4X"
C       READ(ICH, 8226, END=1779, ERR=1779) INST
C8226   FORMAT(41X, 3A2)
        READ(ICH, 8210, END=1779, ERR=1779) (ITXT(I), I=1, 40)
        CALL MAPST(ITXT, 42, 47, 50, INST, 3, 1, $1720)
1720    READ(ICH, 8210, END=1779, ERR=1779) (ITXT(I), I=1, 40)
C       READ(ICH, 8240, END=1779, ERR=1779) (IANLST(I), I=1, 4),
C     1     (IACCNT(I), I=1, 4)
        DO 1730 I=1, 4
C           ISUBM(I)=ITXT(7+I)
            IANLST(I)=ITXT(19+I)
            IACCNT(I)=ITXT(33+I)
1730    CONTINUE

C       LOOK FOR $$$
C
1740    READ(ICH, 8210, END=1779, ERR=1779) (ITXT(I), I=1, 40)
        IP1=NPOS(LOOK2, 2, ITXT, 10, 1)          ;TRICK !!!
        IF(IP1. LE. 0) GOTO 1740

C       READ LOG FILE PARAMETERS
C
        CALL NULST(IPMTH, 1, 4)
        CALL NULST(IEXDT, 1, 4)
        CALL NULST(ICKST, 1, 5)
        CALL NULST(IRGBK, 1, 5)
        CALL NULST(IPVTH, 1, 4)
        CALL NULST(ICMNT, 1, 32)
        CALL NULST(ITMP, 1, 20)
        READ(ICH, 8260, END=1779, ERR=1779) IVS, VOL, VEX, IEXDT, VINJ
        READ(ICH, 8300, END=1779, ERR=1779) IPMTH, ICKST, IRGBK, IPVTH
        READ(ICH, 8320, END=1779, ERR=1779) ICMNT
        IVS(2)=0

        ISOURCE=(NPOS(IVS, 2, MATCH, 6, 1)+2)/3
        IF(ISOURCE. LE. 0) GOTO 1779
        I1=(ISOURCE-1)*6+1
        I2=ISOURCE*6
        I3=ISOURCE
        CALL MAPST(MUNITS, I1, I2, 20, KUNITS, 3, 1, $1610)
        GOTO 1790
```

```
1779    KO=0
        IER=777K
        WRITE(TTO, 8260)
1790    CALL CLOSE(ICH, $1800)
1800    IF(KO. EQ. 0) GOTO 1610

C       CHECK VOLUMES
        IF(VOL. LE. 0. 0. OR. VEX. LE. 0. 0. OR. VINJ. LE. 0. 0) GOTO 1610
        FACTOR=VEX/(VINJ*VOL)

C       PREPARE TO PROCESS QUAN FILE
C
        DO 2012 I=1, MAXN
            SDATA(I, 1)=0.
            SDATA(I, 2)=0.
            IDATA(I, 1)=0
            IDATA(I, 2)=0
2012    CONTINUE

        IFLDSC(10)="QN"
        IER=777K
C       ATTACH QN TO FILENAME AND OPEN
C
        CALL MAPST(IEXTN, 1, 3, 2, IFLFN, 10, LNN, $2479)
        CALL OPENR(IFLFN, ICH, IER, $2479)

C       LOOK FOR AMOUNT AND STORE COMP STRING INTO ICOMP
C
C2070   READ(ICH, 8210, END=2479, ERR=2479) (ITXT(I), I=1, 40)
C       IP1=NPOS(LOOK4, 3, ITXT, 10, 1)
C       IF(IP1. LE. 0) GOTO 2070
C       CALL GOBBL(ITXT, 50, LTX)
C       LTXP=LTX+1
C       IER=777K
C       CALL MOVBT(ICRGS, ITXT, 50, LTXP, $2479)
C       CALL MOVST(ITXT, 1, LTXP, 50, ICOMP, 64, -1, $2479)
C2080   READ(ICH, 8210, END=2479, ERR=2479) (ITXT(I), I=1, 40)
C       CALL GOBBL(ITXT, 50, LTX)
C       IF(LTX. LE. 0) GOTO 2090
C       CALL MOVST(ITXT, 1, LTX, 50, ICOMP, 64, 0, $2479)
C       GOTO 2080

C       LOOK FOR NO  NAME
C
2090    READ(ICH, 8210, END=2479, ERR=2479) (ITXT(I), I=1, 40)
        IP1=NPOS(LOOK5, 4, ITXT, 10, 1)
        IF(IP1. LE. 0) GOTO 2090

C       COUNT NAMES
C
        INM=0
        CALL OPENR(IGCFN, DSK, IER, $2190)
        READ(DSK, 8330, END=2190, ERR=2190) (ITMP(I), I=1, 7),
     1      NM, NO, NN, (ITXT(I), I=1, 16)
2100    READ(DSK, 8350, END=2200, ERR=2190) INDG, (NAME(I), I=1, 35)
        READ(DSK, 8210, END=2190, ERR=2190) (ITXT(I), I=1, 40)
        READ(DSK, 8210, END=2190, ERR=2190) (ITXT(I), I=1, 40)
        READ(ICH, 8360, END=2190, ERR=2190) IND, (ITXT(I), I=1, 35)
        IF(IND. LE. 0) GOTO 2190
        INM=INM+1
        IF(INDG. NE. IND) GOTO 2190
        IF(IND. NE. INM) GOTO 2190
        DO 2120 I=1, 10
            IF(NAME(I). NE. ITXT(I)) GOTO 2190
2120    CONTINUE
        GOTO 2100

2190    WRITE(TTO, 8370)
        CALL CLOSE(DSK, $2479)
        GOTO 2479
```

```
2200    CALL CLOSE(DSK,$2479)
        NIND=INM

C       LOOK FOR NO  M/E
C
2202    NIND=INM
        READ(ICH,8210,END=2479,ERR=2479) (ITXT(I),I=1,40)
        IP1=NPOS(LOOK6,4,ITXT,10,1)
        IF(IP1.LE.0) GOTO 2202

C       READ QUAN DATA AND INSERT IN ARRAYS
C
2210    READ(ICH,8380,END=2479,ERR=2479) IND,(ITMP(I),I=1,9),IREF,RRT,
      1 AR,AM,(ITXT(I),I=1,7)
C       CHECK FOR 'NOT FOUND'
        IP1=NPOS(LOOK3,5,ITMP,9,1)
        IF(IP1.GT.0) GOTO 2230
        IDATA(IND,1)=IREF
        IDATA(IND,2)=1000*(RRT+D4)
        SDATA(IND,1)=AR+D1
        SDATA(IND,2)=AM
2230    IF(IND.LT.NIND) GOTO 2210

C       IF STANDARD THEN GET THE LIBRARY AMOUNTS
C
        IF(ICODE.NE.7) GOTO 2500
2260    READ(ICH,8210,END=2479,ERR=2479) (ITXT(I),I=1,40)
        IP1=NPOS(LOOK7,4,ITXT,10,1)
        IF(IP1.LE.0) GOTO 2260

C       FOUND 'NO  RET(' NOW UPDATE AMOUNTS FOR STANDARDS
2280    READ(ICH,8390,END=2479,ERR=2479) IND,(ITMP(I),I=1,17),AM,
      1 (ITXT(I),I=1,12)
        SDATA(IND,2)=AM
        IF(IND.LT.NIND) GOTO 2280

        GOTO 2500

2479    KO=0
        WRITE(TTO,8400)
2500    CALL CLOSE(ICH,$2510)
2510    IF(KO.EQ.0) GOTO 1610

C####################################################################VVVV
C       CREATE REDUCED DATA FILE AND WRITE ROUNDED DATA
C
        IFLDSC(10)="RD"
        IER=777K
        CALL MAPST(IEXTR,1,3,2,IFLFN,10,LNN,$2640)
        CALL OPENW(IFLFN,LCH,IER,$2640)
        GOTO 2660

2640    WRITE(TTO,8440)
        GOTO 1610

2660    WRITE(LCH,8460) NIND,IACCNT,IPMTH,IANLST,IEXDT,IACDT,IPRDT,ICODE
        X1=VOL+D4
        X2=VEX+D4
        X3=VINJ+D4
        X4=FACTOR+D4
        WRITE(LCH,8480) INST,KUNITS,X1,X2,X3,X4,NCOMP,(IGCDSC(I),I=4,9)
        WRITE(LCH,8500) IRGBK,ICKST,IPVTH
        WRITE(LCH,8520) (ISMPLE(I),I=1,32)
        WRITE(LCH,8520) (ICOND(I),I=1,32)
        WRITE(LCH,8520) (ICMNT(I),I=1,32)

        F=FACTOR
        DO 2730 I=1,NIND
            CONC=SDATA(I,2)*F+D4
        WRITE(LCH,8540) I,CONC,SDATA(I,1),IDATA(I,1),IDATA(I,2)
2730    CONTINUE
        CALL CLOSE(LCH,$2740)
C###################################################################^^^^
```

```
2740    IF(ICODE.NE.0) GOTO 4000

C       PROCESS POSSIBLE DUPLICATE AND/OR SPIKED UNKNOWN
C
        DO 3520 ICODE=1,2
             IF(ICODE.NE.1) GOTO 2800
             ICRDF="D<0>"
             GOTO 2820

2800         ICRDF="F<0>"

2820         IER=777K
             CALL MOVBT(ICRDF,IFLDSC,11,LOC,$2900)
             CALL MOVBT(ICRDF,IFLFN,10,LN,$2900)

             IFLDSC(10)="GN"
             IER=22K
             CALL FSLKU(IFLDSC,$2850,$9000,IER)
             GOTO 2870

2850         WRITE(TTO,8160) IFLDSC(1),(IFLDSC(I),I=4,9),IFLDSC(10),IER
             GOTO 3520

2870         IFLDSC(10)="LG"
             CALL FSLKU(IFLDSC,$2850,$9000,IER)
             IER=777K
             CALL MAPST(IEXTG,1,3,2,IFLFN,10,LNN,$2900)
             CALL OPENR(IFLFN,ICH,IER,$2900)
             IF(ICODE.EQ.1) WRITE(TTO,8170)
             IF(ICODE.EQ.2) WRITE(TTO,8174)
             GOTO 2920

2900         WRITE(TTO,8180) IER
             GOTO 2850

2920         KO=1
             VOL=0.
             VEX=0.
             VINJ=0.
             VX=0.
             IPR=0K
             CALL NULST(ITXT,1,50)

C       LOOK FOR ACQUIRE
C
2930         READ(ICH,8210,END=2969,ERR=2969) (ITXT(I),I=1,40)
             IP1=NPOS(LOOK1,4,ITXT,10,1)
             IF(IP1.LE.0) GOTO 2930

             READ(ICH,8200,END=2969,ERR=2969) (IACDT2(I),I=1,4)

C            LOOK FOR $$$
C
2940         READ(ICH,8210,END=2969,ERR=2969) (ITXT(I),I=1,40)
             IPR=NPOS(LOOK2,2,ITXT,10,1)
             IF(IPR.LE.0) GOTO 2940

C       READ LOG FILE PARAMETERS
C
             CALL NULST(IEXDT2,1,4)
             CALL NULST(KUNITS,1,3)
             CALL NULST(ITMP,1,20)
             READ(ICH,8280,END=2969,ERR=2969) IVS,VOL,VEX,IEXDT2,VINJ
             GOTO 2970
```

```
2969          KO=0
              WRITE(TTO, 8260)
2970          CALL CLOSE(ICH, $2980)
2980          IF(KO. NE. 1) GOTO 2850

              IF(VOL. LE. 0. 0. OR. VEX. LE. 0. 0. OR. VINJ. LE. 0. 0) GOTO 2850
              FACTOR=VEX/(VINJ*VOL)

              DO 3160 I=1, MAXN
                  SDATA(I, 1)=0. 0
                  SDATA(I, 2)=0. 0
                  IDATA(I, 1)=0
                  IDATA(I, 2)=0
3160          CONTINUE

              IFLDSC(10)="GN"
              IER=777K
              CALL MAPST(IEXTN, 1, 3, 2, IFLFN, 10, LNN, $3200)
              CALL OPENR(IFLFN, ICH, IER, $3200)
              GOTO 3210

3200          WRITE(TTO, 8340) IER
              GOTO 2850

C        LOOK FOR NO  M/E
C
3210          READ(ICH, 8210, END=3279, ERR=3279) (ITXT(I), I=1, 40)
              IP1=NPOS(LOOK6, 4, ITXT, 10, 1)                    ; TRICK
              IF(IP1. LE. 0) GOTO 3210

C        READ QUAN DATA AND STORE IN ARRAY
C
3220          READ(ICH, 8380, END=3279, ERR=3279) IND, (ITMP(I), I=1, 9), IREF, RRT,
     1        AR, AM, (ITXT(I), I=1, 7)
              IP1=NPOS(LOOK3, 5, ITMP, 9, 1)
              IF(IP1. GT. 0) GOTO 3240
              IDATA(IND, 1)=IREF
              IDATA(IND, 2)=1000*(RRT+D4)
              SDATA(IND, 1)=AR+D1
              SDATA(IND, 2)=AM
3240          IF(IND. LT. NIND) GOTO 3220
              GOTO 3300

3279          KO=0
              WRITE(TTO, 8400)
3300          CALL CLOSE(ICH, $3310)
3310          IF(KO. EQ. 0) GOTO 2850

C#####################################################################VVVV
C        APPEND RESULTS TO THE REDUCED DATA FILE
C
              IFLDSC(10)="RD"
              IER=777K
              CALL MAPST(IEXTR, 1, 3, 2, IFLFN, 10, LNN, $3489)
              CALL MOVBT(ICRU, IFLDSC, 11, LOC, $3489)
              CALL MOVBT(ICRU, IFLFN, 10, LN, $3489)
              CALL OPEN(IFLFN, LCH, IER, $3489)
              IER=77K
              READ(LCH, 8580, END=3489, ERR=3489) NPIND
              IF(NPIND. NE. NIND) GOTO 3489
              CALL POSITION(LCH, 1, -1, -1, IER, $3340)

C        WRITE SUB HEADER
C
3340          X1=VOL+D4
              X2=VEX+D4
              X3=VINJ+D4
              X4=FACTOR+D4
              WRITE(LCH, 8620) ICODE, IEXDT2, IACDT2, IPRDT, X1, X2, X3, X4
```

```
C       WRITE RESULTS
C
            F=FACTOR
            DO 3480 I=1,NIND
                CONC=SDATA(I,2)*F+D4
                WRITE(LCH,8540) I,CONC,SDATA(I,1),IDATA(I,1),IDATA(I,2)
3480        CONTINUE

C           WRITE NEW ICODE IN REDUCED DATA FILE
C
            CALL POSITION(LCH,0,0,52,IER,$3489)
            READ(LCH,8580,END=3489,ERR=3489) ICD
            CALL POSITION(LCH,0,0,52,IER,$3489)
            ICD=ICD+ICODE
            WRITE(LCH,8600) ICD
            GOTO 3490

3489        WRITE(TTO,8660) IFLDSC(1),(IFLDSC(I),I=4,9),IFLDSC(10),IER
3490        CALL POSITION(LCH,1,-1,-1,IER,$3500)
3500        CALL CLOSE(LCH,$3520)
C#################################################################^^^^
3520    CONTINUE

3600    IER=77K
        CALL MOVBT(ICRU,IFLDSC,11,LOC,$9000)
        CALL MOVBT(ICRU,IFLFN,10,LN,$9000)
C
C       HERE'S WHERE ALL THE QUAL GOODIES GO
C
4000    IFLDSC(10)="GA"
        CALL FSLKU(IFLDSC,$6000,$6000,IER)
        IFLDSC(10)="RD"
        CALL MAPST(IEXTR,1,3,2,IFLFN,10,LNN,$5000)
        CALL OPEN(IFLFN,LCH,IER,$9000)
        CALL POSITION(LCH,1,-1,-1,IER,$4040)
        GOTO 5000

4040    WRITE(MCH,8680)
        CALL MAPST(IEXTA,1,3,2,IFLFN,10,LNN,$5000)
        CALL OPENR(IFLFN,ICH,IER,$5000)
        CALL POSITION(ICH,0,0,112,IER,$5000)
        CALL FCOPY(ICH,LCH,$5000,IE,IER)
        CALL CLOSE(ICH,IE)
        CALL CLOSE(LCH,IE)
        GOTO 6000

5000    CALL RESET
        WRITE(TTC,8700) IER

6000    IF(LSTMD.EQ."L<0>") GOTO 1500
```

```
7000    CALL LOADIT("DMPG2")

8000    FORMAT("1",/," DATA REDUCTION MODULE (DMPG1)",/)
8010    FORMAT(" GC/METHOD TEMPLATE NAME [",4A2,"]  > ",Z)
8020    FORMAT(" <7>GC<57>MT ACCESS ERROR: ",OI6)
8060    FORMAT(" NAME LIST (L) OR MANUAL NAME (M) ENTRY [",A2,"] > ",Z)
8080    FORMAT(" NAME OF LIST FILE [",8A2,"] > ",Z)
8100    FORMAT(" <7>NAMELIST ACCESS ERROR: ",OI6)
8120    FORMAT(" SAMPLE NAME ($&)  [",8A2,"] > ",Z)
8140    FORMAT(" <7>DUPLICATE OR SPIKE SAMPLE ENCOUNTERED!",
     1      " SAMPLE WILL ONLY BE PROCESSED IN CONJUNCTION",
     1      " WITH AN UNKNOWN SAMPLE.")
8150    FORMAT(" NOW PROCESSING PRIMARY SAMPLE: ",6A2)
8160    FORMAT(" NO DATA FOR: ",A2,":",6A2,".",A2,"  ERROR: ",OI6)
8170    FORMAT(" NOW PROCESSING DUPLICATE UNKNOWN")
8174    FORMAT(" NOW PROCESSING SPIKED UNKNOWN")
8180    FORMAT(" <7>FILE ERROR: ",OI6,"  ",Z)
8200    FORMAT(4A2)
8210    FORMAT(40A2)
8220    FORMAT(8X,32A2,8X)
8260    FORMAT(" ERROR READING LGFILE!")
8280    FORMAT(2A2,F10.3,6X,F10.3,6X,4A2,6X,F10.3)
8300    FORMAT(4X,4A2,6X,5A2,6X,5A2,6X,4A2)
8320    FORMAT(4X,32A2)
8330    FORMAT(7A2,3I4,16A2,1X)
8340    FORMAT("  <7>GN OPEN ERROR: "OI4)
8350    FORMAT(I3,1X,35A2,2X)
8360    FORMAT(I3,2X,35A2)
8370    FORMAT(" <7>NAME COMPARISON PROBLEM WITH GC<57>MT FILE")
8380    FORMAT(I3,1X,9A2,I3,F7.3,6X,F12.0,F11.3,1X,7A2)
8390    FORMAT(I3,2X,17A2,F10.2,12A2)
8400    FORMAT(" <7>ERROR READING QUANFILE!")
8440    FORMAT(" <7>RD OPEN ERROR : ",Z)
8460    FORMAT(1X,I4,4A2,4A2,4A2,4A2,4A2,4A2,I4)
8480    FORMAT(1X,3A2,3A2,4F8.3,2A2,6A2)
8500    FORMAT(1X,5A2,5A2,4A2,8X,"RESULTS FILENAME  MM/DD/YY")
8520    FORMAT(1X,32A2)
8540    FORMAT(1X,I4,F12.3,F12.0,I5,I6)
8580    FORMAT(I4)
8600    FORMAT(1X,I4)
8620    FORMAT(1X,I4,4A2,4A2,4A2,4F8.3)
8660    FORMAT(" <7>CAN'T APPEND TO: ",A2,":",6A2,".",A2,"   ERROR: ",OI6)
8680    FORMAT("   9 QUALITATIVE IDENTIFICATION SEGMENT")
8700    FORMAT(" <7>QUAL FILE ACCESS ERROR: ",OI6)

9000    WRITE(TTO,9010) IER
9010    FORMAT(" <7>ERROR FATAL: ",OI4)
9020    CALL RESET
        CALL LOADIT("EXEC")
        END
```

```
C       DMPG2.FR     DATA MANAGEMENT REPORT GENERATOR
C       D.RYAN 9/81      LAST REVISION: 09/23/82

        PARAMETER TTO=10,LPT=2,DSK=1
        PARAMETER MAXN=120,MAXSM=25

        COMMON /CJNK1/
     1  IRPDT(4),KVOL(2),IRRNM(9),IRRDT(4),
     1  INST(3),ISMPLE(32),IANLST(4),ICOND(32),IACCNT(4),
     1  ITMP(20),ITXT(50),IACDT(4),IACDT2(4),KUNITS(3),
     1  IRUN(10),NAME(35),METH(2),IPMTH(4),IPVTH(4),
     1  IEXDT(4),IEXDT2(4),ICKST(5),IRGBK(5),ICMNT(32),
     1  NCOMP(2),ICOMP(64),ICAS(6),IDATA(MAXN,2),SDATA(MAXN,2),IVS(2),
     1  ICRU,ICRGS,ICREQ,NIND,ISAMP,IYR,IMO,IDAY,IHR,IMIN,ISEC,IX,INM,
     1  IC,IEE,IER,NSAMP,LTYP,LOC,ICODE,ICRDF,NPIND,ILEN,IANS,LNN,LN,
     1  IPOS,IPR,ILINE,IP1,IP2,IP3,LP,LP1,LTX,I,I1,I2,I3,IND,IREF,KO,KP1,
     1  RRT,VOL,VEX,VINJ,VX,X,FACTOR,F,CONC,D1,D2,D3,D4,X1,X2,X3,X4,
     1  AR,AM,ICH,LCH,MCH,ISOURCE

        COMMON /CJNK2/
     1  MUNITS(20),MATCH(6),MCODE(4),LOOK1(4),LOOK2(2),LOOK3(5),LOOK4(3),
     1  LOOK5(4),LOOK6(4);LOOK7(4),IEXTC(2),IEXTN(2),IEXTR(2),IEXTG(2),
     1  IEXTA(2)

        COMMON /CJNK3/
     1  IQCDSC(11),IQCFN(10),IFLDSC(11),IFLNM(8),IFLFN(10),
     1  INLDSC(11),INLNM(8),IS(11,MAXSM),LSTMD

        COMMON /CJNK4/
     1  IQCNM(4),IPRDT(4)

        COMMON /CMAIN/
     1  NPROG,IDFSW,IDFSW1,ITEMP(15),
     1  IDFILE(12,6),ISFILE(11,4),
     1  INSTD(3),IREFTABLE(11),
     1  IADD1,IADD2,ISUB1,ISUB2,ISUB3,ISUB4,
     1  JADD1,LIBRNAME,LIBRNUMB,IBUF,
     1  ISCN(45),IACQ(100),IDAT(45)

        DATA MUNITS/"UG","<57>L","  ","UG","<57>K","GD","UG","<57>K",
     1  "GW","UG","<57>C","M ",0,0,0,0,0,0,0,0/
        DATA MATCH/"VS","<75>M","D<75>","MW","<75>A","S<75>"/
        DATA MCODE/"UD","FX","CR","NS"/
        DATA LOOK1/"AC","QU","IR","E<0>"/
        DATA LOOK2/"$$$<0>"/
        DATA LOOK3/"NO","T ","FO","UN","D<0>"/
        DATA LOOK4/"AM","OU","NT"/
        DATA LOOK5/"NO","  ","NA","ME"/
        DATA LOOK6/"NO","  ","M<57>","E "/
        DATA LOOK7/"NO","  ","RE","T("/
        DATA IEXTC/".Q","C<0>"/
        DATA IEXTN/".Q","N<0>"/
        DATA IEXTG/".L","G<0>"/
        DATA IEXTR/".R","D<0>"/

        D1=0.5
        D2=0.05
        D3=0.005
        D4=0.0005
        ICRU="U<0>"
        ICRGS="?<0>"
        ICREQ="=<0>"
        ICRDF="N<0>"
        KCHAN=1
        NIND=0
        KO=0
        NSAMP=0
        ICH=3
        LCH=4
        MCH=5
        IFROM=1
        CALL NULST(ITMP,1,20)
        CALL NULST(ITXT,1,50)
        CALL GETTI(IYR, IMO, IDAY, IHR, IMIN, ISEC, $50, IX, IX, IX, IX, IX, IX)
        IRPDT(1)=(IMO/10+48)*256+MOD(IMO, 10)+48
        IRPDT(2)=47*256+IDAY/10+48
        IRPDT(3)=(MOD(IDAY, 10)+48)*256+47
        IRPDT(4)=(MOD(IYR/10, 10)+48)*256+MOD(IYR, 10)+48
        GOTO 140

50      IRPDT(1)="MM"
        IRPDT(2)="<57>D"
        IRPDT(3)="D<57>"
        IRPDT(4)="YY"

C       SALUTATION
C
140     WRITE(TTO, 8000)

        IF(LSTMD. EQ. "L<0>") GOTO 1260
        IF(LSTMD. EQ. "M<0>") GOTO 4200
        IFROM=0

C       GET MODE: MANUAL OR NAMELIST
1150    WRITE(TTO) " <7>"
        IANS="M<0>"
        CALL INPTS(LSTMD, IANS, 1, $8060, $7300)
        IF(IANS. NE. "L<0>". AND. IANS. NE. "M<0>") GOTO 1150

        IF(LSTMD. NE. "M<0>") GOTO 1200

C       GET NEXT MANUALLY ENTERED NAME
C
        DO 1180 I=1, 6
            IFLNM(I)=ISFILE(I+3, 1)
1180    CONTINUE
        CALL INPTS(ITXT, IFLNM, 8, $8120, $7300)
        GOTO 1590

C       GET NAMELIST AND READ
C
1200    CALL NULST(INLNM, 1, 8)
        DO 1210 I=1, 6
            INLNM(I)=IDFILE(I+3, 1)
1210    CONTINUE
        CALL INPTS(ITMP, INLNM, 8, $8080, $7300)
        DO 1230 I=4, 9
            INLDSC(I)=INLNM(I-3)
1230    CONTINUE
        INLDSC(1)="D<0>"
        INLDSC(2)=0
        INLDSC(3)=1
        INLDSC(10)="NL"
        INLDSC(11)=0
        CALL FSLKU(INLDSC, $1240, $9000, IEE)
        GOTO 1260

C       NAMELIST DOES NOT COMPUTE
1240    WRITE(TTO, 8100) IEE
        GOTO 1200

1260    IC=0
        CALL FMOPS(INLDSC, KCHAN, $1240, IER)
        CALL FMRD(IS, KCHAN, 11*MAXSM, $1270, IEE)
        IEE=0
1270    CALL FMCLS(KCHAN, $1280)
1280    IF(IEE. GT. 0) GOTO 1240
        IC=(11*MAXSM+IEE)/11
        NSAMP=IC
        ISAMP=0

C       GET NEXT NAME FROM LIST
C
1500    CALL NULST(IFLNM, 1, 8)
        ISAMP=ISAMP+1
        IF(ISAMP. GT. NSAMP) GOTO 7300
        DO 1520 I=4, 9
            I1=I-3
            IFLNM(I1)=IS(I, ISAMP)
1520    CONTINUE
```

```
C       PREPARE FILENAME FOR PROCESSING
C
1590    CALL NULST(IFLFN, 1, 10)
        DO 1600 I=1, 6
             I1=I+3
             IFLDSC(I1)=IFLNM(I)
             IFLFN(I)=IFLNM(I)
1600    CONTINUE
C       FIND KEY CHARACTER AND SET SAMPLE TYPE CODE
C
        LN=LENG(IFLNM, 8)
        LNN=LN+1
        IFLDSC(10)="RD"
        CALL MAPST(IEXTR, 1, 3, 2, IFLFN, 10, LNN, $4210)

4200    CALL OPENR(IFLFN, ICH, IER, $4210)
        GOTO 4220

C       HANDLE REDUCED DATA FILE I/O ERROR
C
4210    CALL RESET
        WRITE(TTO, 8200) (IFLDSC(I), I=4, 9)
        GOTO 7200

C       READ HEADER PARAMETERS
C
4220    READ(ICH, 8230, END=4210, ERR=4210) NIND, IACCNT, IPMTH, IANLST,
     1       IEXDT, IACDT, IPRDT, ICODE
        READ(ICH, 8240, END=4210, ERR=4210) INST, KUNITS, VOL, VEX, VINJ, FACTOR,
     1      NCOMP, (ITMP(I), I=1, 6)
        READ(ICH, 8250, END=4210, ERR=4210) IRGBK, ICKST, IPVTH, IRRNM, IRRDT
        READ(ICH, 8260, END=4210, ERR=4210) ISMPLE
        READ(ICH, 8260, END=4210, ERR=4210) ICOND
        READ(ICH, 8260, END=4210, ERR=4210) ICMNT

C       PREPARE HEADER VARIABLES AND GET GC/MT FILENAME
C
        CALL GOBBL(ITMP, 20, LTX)
        DO 4240 I=1, 6
             IGCFN(I)=ITMP(I)
             IGCDSC(I+3)=ITMP(I)
4240    CONTINUE
        CALL MAPST(IEXTC, 1, 3, 2, IGCFN, 10, 8, $4260)
        IGCDSC(1)="D<0>"
        IGCDSC(2)=0
        IGCDSC(3)=1
        IGCDSC(10)="GC"
        IGCDSC(11)=0
        CALL FSLKU(IGCDSC, $4260, $9000, IER)
        CALL MAPST(KUNITS, 4, 6, 3, KVOL, 2, 1, $4260)
        ICP=ICODE+1
        ITYPE=NCHAR(MCODE, 4, ICP)
        IF(ICP.LE.4) ICP=1
        ICHK=0
        X1=VOL+D4
        X2=VEX+D4
        X3=VINJ+D4
        X4=FACTOR+D4

        CALL OPENR(IGCFN, LCH, IER, $4260)
        READ(LCH, 8270, END=4260, ERR=4260) (ITMP(I), I=1, 7), NM, NO, NN,
     1      (ITXT(I), I=1, 16)
        IF(NM.NE.NIND) GOTO 4260
        GOTO 4280

C       GC/MT ACCESS ERROR
C
4260    CALL RESET
        WRITE(TTO, 8280)
        GOTO 7200
```

```
4280    CALL DELETE("1:DMPRPTG.99",IER,$4290)
4290    CALL OPENW("1:DMPRPTG.99",MCH,IER,$9000)
        WRITE(MCH,8300) (IFLNM(I),I=1,6),IPRDT

        WRITE(TTO,8140) (IFLDSC(I),I=4,9)
        WRITE(MCH,8310) IACCNT,ITYPE
        WRITE(MCH,8320) KUNITS,(IGCDSC(I),I=4,7)
        WRITE(MCH,8330) X1,KVOL,X4
        WRITE(MCH,8340) IEXDT,IACDT
        WRITE(MCH,8350) X2,X3
        WRITE(MCH,8360) IANLST,INST
        WRITE(MCH,8370) IRGBK,ICKST

C                U    D    F    X    C    R    N    S
        GOTO (4500,4500,4500,4500,4510,4520,4500,4530) ICP

C       SUB HEADER FOR U,D,F,X,N
C
4500    WRITE(MCH,8510)
        WRITE(MCH,8520)
        GOTO 4535

C       SUB HEADER FOR C
C
4510    WRITE(MCH,8740)
        WRITE(MCH,8750)
        GOTO 4535

C       SUB HEADER FOR R
C
4520    WRITE(MCH,8780)
        WRITE(MCH,8790)
        GOTO 4535

C       SUB HEADER FOR S
C
4530    WRITE(MCH,8820)
        WRITE(MCH,8830)
4535    WRITE(MCH,8530)

C       READ GC/MT AND RD LINE ITEMS
C
4540    READ(LCH,8540,END=4260,ERR=4260) INDG,(NAME(I),I=1,35)
        READ(LCH,8550,END=4260,ERR=4260) (ICAS(I),I=1,6),IUSE,RMDL,
     1      XMDL,A,B,C,D,KEYD
        READ(LCH,8560,END=4260,ERR=4260) RMEAN,SIGR,KEYR,XLSA2,SMEAN,SIGS,
     1      KEYS,XLCT,CMEAN,SIGC,KEYC
        CALL GOBBL(NAME,35,LTX)
        READ(ICH,8570,END=4210,ERR=4210) IPMCT,CONC,AR,IDT,IRRT
        LIMIT="  "
        IF(X5.LT.A) LIMIT="<74> "
        IF(X5.GT.B) LIMIT="<76> "
        X5=CONC+D4
        X6=AR+D1
C       SKIP INTERNAL STANDARDS UNLESS SAMPLE IS A STANDARD
        IF((IUSE.EQ.3).AND.(ICODE.NE.7)) GOTO 4590
        IF(INDG.NE.IPMCT) GOTO 7200

C                U    D    F    X    C    R    N    S
        GOTO (4560,4910,4740,9000,5230,5530,4560,6030) ICP
C  ICODE=        0    1    2    3    4    5    6    7

C       OUTPUT UNKNOWN AND NON-SFC SAMPLE INFO
C
4560    IF(IUSE.EQ.2) GOTO 4600
        SDATA(INDG,2)=CONC
        WRITE(MCH,8580) INDG,(NAME(I),I=1,12),X5,LIMIT,ICAS,X6
        IF(LTX.LE.24) GOTO 4590
        WRITE(MCH,8585) (NAME(I),I=13,24)

C       U (D,F) AND N LINE TERMINATOR
C
4580    IF(LTX.LE.48) GOTO 4590
        WRITE(MCH,8590) (NAME(I),I=25,35)
4590    IF(INDG.EQ.NIND) GOTO 4700
        GOTO 4540
```

```
C       OUTPUT SURROGATE INFO FOR U OR N
C
4600    IF(XLSA2.LE.D2) GOTO 4590
        CALL QCTST(CONC,SMEAN,SIGS,XLSA2,IGCMS,RECOV)
        X7=XLSA2+D3
        X8=RECOV+D2
        WRITE(MCH,8610)
        WRITE(MCH,8620) INDG,(NAME(I),I=1,12),X8,X5
        WRITE(MCH,8630) (NAME(I),I=13,24),ICAS,X7,IGCMS,KEYS
        IF(LTX.LE.48) GOTO 4650
        WRITE(MCH,8590) (NAME(I),I=25,35)
4650    WRITE(MCH,8610)
        GOTO 4590

C       OUTPUT D OR F HEADER
C
4700    IF((ICODE.EQ.0).OR.(ICODE.EQ.6)) GOTO 6500
        IF(ICHK.EQ.ICODE) GOTO 6500
        CALL POSITION(LCH,0,0,0,IER,$9000)
        READ(ICH,8640,END=4210,ERR=4210) ICD,IEXDT2,IACDT2,IPRDT,VOL,
     1  VEX,VINJ,FACTOR
        ICHK=ICHK+ICD
        ICP=ICD+1
        X1=VOL+D4
        X2=VEX+D4
        X3=VINJ+D4
        X4=FACTOR+D4
        IF((ICD.NE.1).AND.(ICD.NE.2)) GOTO 4260
        READ(LCH,8270,END=4260,ERR=4260) (ITMP(I),I=1,7),NM,NO,NN,
     1      (ITXT(I),I=1,16)
        IF(ICD.EQ.1) WRITE(MCH,8690) IPRDT
        IF(ICD.EQ.1) WRITE(TTO,8160)
        IF(ICD.EQ.2) WRITE(MCH,8650) IPRDT
        IF(ICD.EQ.2) WRITE(TTO,8180)
        WRITE(MCH,8330) X1,KVOL,X4
        WRITE(MCH,8340) IEXDT2,IACDT2
        WRITE(MCH,8350) X2,X3
        IF(ICD.EQ.1) WRITE(MCH,8700)
        IF(ICD.EQ.2) WRITE(MCH,8660)
        GOTO 4535

C       OUTPUT SPIKE INFO
C
4740    IF(IUSE.EQ.2) GOTO 4600
        IF(XLSA2.LE.D2) GOTO 4590
        X=CONC-SDATA(INDG,2)
        CALL QCTST(X,SMEAN,SIGS,XLSA2,IGCMS,RECOV)
        X8=RECOV+D2
        X7=SDATA(INDG,2)+D4
        WRITE(MCH,8670) INDG,(NAME(I),I=1,12),X8,X7,IGCMS,KEYS
        X7=XLSA2+D3
        WRITE(MCH,8680) (NAME(I),I=13,24),X7,X5
        GOTO 4580

C       OUTPUT DUPLICATE INFO
C
4910    IF(IUSE.EQ.2) GOTO 4600
        IF(CONC.LE.D4.AND.SDATA(INDG,2).LE.D4) GOTO 4590
        CONCO=SDATA(INDG,2)
        XBAR=(CONC+CONCO)/2
        DIFF=CONC-CONCO
        IF(DIFF.LT.0.0) DIFF=-DIFF
        RC=D+C*XBAR
        IGCMS="NO"
        IF(DIFF.LE.RC) IGCMS="OK"
        X7=CONCO+D4
        X8=XBAR+D4
        WRITE(MCH,8710) INDG,(NAME(I),I=1,12),X7,X8,IGCMS,KEYD
        X7=DIFF+D4
        WRITE(MCH,8720) (NAME(I),I=13,24),X5,X7
        IF((CONCO.LT.RMDL).OR.(CONC.LT.RMDL)) GOTO 4580
        PRR=100.0*DIFF/XBAR+D3
        WRITE(MCH,8730) (NAME(I),I=25,35),PRR
        GOTO 4590
```

```
C       OUTPUT CONTROL STANDARD INFO
C
5230    IF(XLCT.LE.D3) GOTO 4590
        CALL QCTST(CONC,CMEAN,SIGC,XLCT,IGCMS,RECOV)
        X7=RECOV+D2
        WRITE(MCH,8760) INDQ,(NAME(I),I=1,12),X7,X5,IGCMS,KEYC
        X7=XLCT+D3
        WRITE(MCH,8770) (NAME(I),I=13,24),ICAS,X7

C       C, R, AND S LINE TERMINATORS
C
5240    IF(LTX.LE.48) GOTO 5260
        WRITE(MCH,8590) (NAME(I),I=25,35)
5260    IF(INDQ.EQ.NIND) GOTO 7000
        GOTO 4540

C       OUTPUT REAGENT BLANK INFO
C
5530    IF(AR.LE.D1) GOTO 4590  ;NOT FOUND
        CALL QCTST(CONC,RMEAN,SIGR,-1.0,IGCMS,RECOV)
        X7=RMEAN+3*SIGR+D3
        X8=RMEAN+2*SIGR+D3
        WRITE(MCH,8800) INDQ,(NAME(I),I=1,12),X5,X7,IGCMS,KEYR
        WRITE(MCH,8810) (NAME(I),I=13,24),ICAS,X8
        GOTO 5240


C       OUTPUT STANDARD INFO
C
6030    LIMIT="  "
        WRITE(MCH,8580) INDQ,(NAME(I),I=1,12),X5,LIMIT,ICAS,X6
        IF(LTX.LE.24) GOTO 5240
        WRITE(MCH,8585) (NAME(I),I=13,24)
        GOTO 5240

C       MANAGE QUALITATIVE OUTPUT HERE
C
6500    READ(ICH,8860,END=7000,ERR=7000) ICD,(ITXT(I),I=1,32)
        IF(ICD.NE.9) GOTO 7000
        WRITE(MCH,8880) (IFLNM(I),I=1,6),IPRDT

6540    READ(ICH,8900,END=7000,ERR=7000) (ITXT(I),I=1,37)
        WRITE(MCH,8920) (ITXT(I),I=1,37)
        GOTO 6540

C       OUTPUT REPORT
C
7000    WRITE(MCH,8940)
        CALL RESET
        KCHAN=LPT
        CALL OPLPT(KCHAN,$7020)
        CALL OPENR("1:DMPRPTQ.99",DSK,IER,$9000)
        CALL FCOPY(DSK,KCHAN,$7020,IE,IER)
7020    CALL CLOSE(DSK,IE)
        CALL CLOSE(KCHAN,IE)

7200    IF(LSTMD.EQ."L<0>") GOTO 1500
7300    CALL RESET
        CALL LOADIT("DMPC")
```

```
8000     FORMAT(" ",/," REPORT GENERATION MODULE (DMPG2)",/)
8060     FORMAT(" NAMELIST (L) OR MANUAL (M) NAME ENTRY [",A2,"] > ",Z)
8080     FORMAT(" NAME OF LIST FILE [",8A2,"] > ",Z)
8100     FORMAT(" <7>NAMELIST ACCESS ERROR: ",O16)
8120     FORMAT(" SAMPLE NAME ($&)  [",8A2,"] > ",Z)
8140     FORMAT(" NOW REPORTING ON PRIMARY SAMPLE: ",6A2)
8160     FORMAT(" NOW REPORTING ON DUPLICATE UNKNOWN")
8180     FORMAT(" NOW REPORTING ON SPIKED UNKNOWN")
8200     FORMAT(" <7>REDUCED DATA FILE ACCESS ERROR FOR ",6A2)
8230     FORMAT(I4,6(4A2),I4)
8240     FORMAT(2(3A2),4F8.3,2A2,6A2)
8250     FORMAT(5A2,5A2,4A2,8X,9A2,4A2)
8260     FORMAT(32A2)
8270     FORMAT(7A2,3I4,16A2,1X)
8280     FORMAT(" <7>GC<57>METHOD TEMPLATE FILE ACCESSS ERROR")
8300     FORMAT(" ",T20,"GS<57>MS COMBINED MEASUREMENT AND"/
      1  T24"QUALITY CONTROL REPORT"/T22,6A2,T40,4A2/)
8310     FORMAT(1X"   ACCOUNT ID: "4A2,T38"SAMPLE TYPE: "A2)
8320     FORMAT(1X"   UNITS: "3A2,T38"GC<57>MT NAME: "4A2)
8330     FORMAT(1X"   SAMPLE VOLUME:      "F8.3" ",2A2,T38,
      1  "FACTOR:             "F8.3)
8340     FORMAT(1X"   EXTRACTED: "4A2,T38"INJECTED: "4A2)
8350     FORMAT(1X"   EXTRACTION VOLUME: "F8.3" ML"T38
      1  "INJECTION VOLUME: "F8.3" UL")
8360     FORMAT(1X"   ANALYST: "4A2,T38"INSTRUMENT: "3A2)
8370     FORMAT(1X"   REAGENT BLANK: "5A2,T38,"CONTROL STANDARD: "5A2/)
8510     FORMAT(1X,T37"MEASURED"T53"CAS")
8520     FORMAT(2X"NO  COMPOUND NAME"T39"CONC"T52"NUMBER"T67"AREA")
8530     FORMAT(" =============================",T36,
      1  "=============================")
8540     FORMAT(I3,1X,35A2,2X)
8550     FORMAT(6A2,1X,I3,1X,2F7.2,1X,4F10.3,A1,1X)
8560     FORMAT(2F7.2,A1,2(F8.2,2F6.1,A1),1X)
8570     FORMAT(I4,F12.3,F12.0,I5,I6)
8580     FORMAT(I4,2X,12A2,T33,F12.3,A2,6A2,T61,F12.0)
8585     FORMAT(1X,T7,12A2)
8590     FORMAT(1X,T6,11A2)
8610     FORMAT(" ---------------------------------",T36,
      1  "---------------------------------")
8620     FORMAT(I4,2X,12A2,T32"%REC= "F7.1" %  LS1="F12.3,T68,"GC=")
8630     FORMAT(" "T7,12A2,T32"CAS "6A2" LS2=   "F8.2,T68,A2,"-",A1)
8640     FORMAT(I4,3(4A2),4F8.3)
8650     FORMAT(1H1,/T24"SPIKED SAMPLE GC REPORT"/1X,T31,4A2/)
8660     FORMAT(1X,/T32"% RECOVERY<57>     MEASURED ORIGINAL   GC"/
      1  2X"NO  COMPOUND NAME"T32"AMOUNT ADDED    AND ",
      1  "SPIKED FINAL  ASSESS")
8670     FORMAT(I4,2X,12A2,T32,"%REC=",F7.1," %  LSO=",F12.3,T68,A2,"-",A1)
8680     FORMAT(1X,T7,12A2,T32,"LSA=",F9.2,"   LSF=",F12.3)
```

```
8690      FORMAT(1H1/T22"DUPLICATE SAMPLE GC REPORT"/1X,T31,4A2,/)
8700      FORMAT(1X,/T35"MEASURED"T55"MEAN<57>"T68"GC"/2X
     1    "NO  COMPOUND NAME"T33"CONCENTRATIONS       DIFFERENCE-   ASSESS")
8710      FORMAT(I4,2X,12A2,T32"LD1="F12.3"   X="F12.3,T68,A2,"-",A1)
8720      FORMAT(" "T7,12A2,T32"LD2="F12.3"   R="F12.3)
8730      FORMAT(1X,T6,11A2,T32"%RR=",F11.2" %")
8740      FORMAT(1X,T34"RECOVERY<57>"T52"MEASURED<57>"T68"GC")
8750      FORMAT(1X"NO  COMPOUND NAME"T36"CAS #"T52
     1    "EXPECTED       ASSESS")
8760      FORMAT(I4,2X,12A2,T32"%REC="F7.1" %     LCM="F12.3,T68,A2,"-",A1)
8770      FORMAT(1X,T7,12A2,T32"CAS "6A2"  LCT=    "F8.2)
8780      FORMAT(1X,T35"MEASURED<57>"T52"CONTROL AND      GC")
8790      FORMAT(1X" NO  COMPOUND NAME"T36"CAS #"T51
     1    "WARNING LIMITS  ASSESS")
8800      FORMAT(I4,2X,12A2,T32"LRB="F12.3"  UCL="F9.2,T68,A2,"-",A1)
8810      FORMAT(1X,T7,12A2,T32"CAS  "6A2" UWL="F9.2)
8820      FORMAT(1X,T39"KNOWN"T53"CAS")
8830      FORMAT(1X"NO  COMPOUND NAME"T39"CONC"T52"NUMBER"T67"AREA")
8860      FORMAT(I3,1X,32A2)
8880      FORMAT(1H1,T26,"GC<57>MS QUALITATIVE AND",/,
     1            T26,"IDENTIFICATION REPORT",/,
     1            T24,6A2,T42,4A2,/)
8900      FORMAT(37A2)
8920      FORMAT(" ",37A2)
8940      FORMAT("1")

9000      WRITE(TTO,9010) IER
9010      FORMAT(" <7>ERROR FATAL: ",OI4)
9020      CALL RESET
          CALL LOADIT("EXEC")

          END
```

```
C       DMPG1.FR  GC/METHOD TEMPLATE EDITOR - NEW TEMPLATE CREATOR
C       D.RYAN 2/82     LAST REVISION: 8/23/83 FOR DEFAULT PARMETH FIX

        PARAMETER TTO=10,LPT=2,ICH=3,LCH=4,MAXM=120

        COMMON /CJNK1/
     1  NAME(36),ICAS(6),ITMP(40),ITXT(70),IGNFN(11),KUNITS(3),
     1  ICINT(3),ICDT(4),IUINT(3),IUDT(4),IARRAY(MAXM),
     1  IYR,IMO,IDAY,IX,KEYD,NHOFF,NRECS,IANS,D1,D2,D3,D4,ILEN,I,IEE,
     1  ILOC,IP1,INMS,NIND,IND,IREF,INM,ISEQ,RMDL,XMDL,A,B,C,D,
     1  RMEAN,SIGR,XLSA2,SMEAN,SIGS,XLCT,CMEAN,SIGC,XTMP,IDEF,
     1  IUSE,X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,Y1,Y2,Y3,Y4,NM,NO,NR,
     1  MOP,ICHN,IOPT,ISBFLD,INSTR,INSTP,ISTRT,ISTOP,NREC,IBIT,
     1  KEYR,KEYS,KEYC,KCHAN,IE,I1,I2

        COMMON /CJNK2/
     1  LOOK5(4),LOOK6(4),LOOK7(5),IEXTC(2),IEXTN(2),IUPT(32),
     1  IGST3(3),NUMBS(5)

        COMMON /CJNK3/
     1  IGCDSC(11),IGCFN(8),INIT(3),IH1,IM1,IS1

        COMMON /CJNK4/
     1  IGCNM(4),IPRDT(4)

        COMMON /CMAIN/ NPROG,IDFSW,IDFSW1,ITEMP(15),
     1  IDFILE(12,6),ISFILE(11,4),INSTD(3),IREFTABLE(11),
     1  IADD1,IADD2,ISUB1,ISUB2,ISUB3,ISUB4,JADD1,
     1  ILIBNAME,ILIBNUMB,IBUF,ISCN(45),IACQ(100),IDAT(45)

        DATA LOOK5/"NO  NAME"/
        DATA LOOK6/"NO  M<57>E "/
        DATA LOOK7/"NOT FOUND<0>"/
        DATA ICAS/" 000000<55>00<55>0"/
        DATA IEXTC/".GC<0>"/
        DATA IEXTN/".GN<0>"/

        CALL GETTI(IYR,IMO,IDAY,IH1,IM1,IS1,*50,IX,IX,IX,IX,IX,IX)
        IPRDT(1)=(IMO/10+48)*256+MOD(IMO,10)+48
        IPRDT(2)=47*256+IDAY/10+48
        IPRDT(3)=(MOD(IDAY,10)+48)*256+47
        IPRDT(4)=(MOD(IYR/10,10)+48)*256+MOD(IYR,10)+48
        GOTO 100

50      WRITE(TTO) " <7>PLEASE SET TIME AT MSDS"
        GOTO 7200

100     KEYD="D<0>"
        NHOFF=60
        NRECS=208
        IANS=0
        D1=0.5
        D2=0.05
        D3=0.005
        D4=0.0005

        WRITE(TTO,8000)

        CALL NULST(IGCNM,8,4)
        CALL NULST(IGCDSC,1,11)
        CALL NULST(ITMP,1,40)
        CALL NULST(ITXT,1,50)
        CALL NULST(IGNFN,1,11)
        CALL NULST(IGCFN,1,8)
        CALL NULST(KUNITS,1,3)
        CALL NULST(IARRAY,1,MAXM)
        CALL NULST(INIT,1,3)
```

```
200     ITMP(1)="IN"
        ITMP(2)="IT"
        ITMP(3)="S<0>"
        CALL INPTS(INIT,ITMP,3,$8020,$7200)
        CALL PADST(INIT,3)

        CALL INPTS(ITMP,IGCNM,4,$8040,$7200)

        ILEN=LENG(IGCNM,4)
        IF(ILEN.NE.7) GOTO 440

        DO 240 I=1,4
            IGCFN(I)=IGCNM(I)
240     CONTINUE
        CALL MAPST(IEXTC,1,3,2,IGCFN,8,8,$440)

        IGCDSC(1)="D<0>"
        IGCDSC(3)=1
        IGCDSC(10)="GC"
        DO 280 I=4,7
            IGCDSC(I)=IGCNM(I-3)
280     CONTINUE
        CALL FSLKU(IGCDSC,$300,$500,IEE)
        GOTO 2900

C       FILE NOT FOUND
C
300     DO 340 I=1,5
            IGCFN(6-I+1)=IGCFN(6-I)
340     CONTINUE
        IGCFN(1)="1:"
C       WRITE(TTO,360) (IGCFN(I),I=1,8),IEE
C360    FORMAT(" GCFILE: ",8A2," ERROR: ",OI6)
        GOTO 600

440     WRITE(TTO) " <7>ERROR ON GC<57>MT NAME.  TRY AGAIN!"
        GOTO 200

500     WRITE(TTO,8060) IEE
        GOTO 200

600     CALL PADST(IGCNM,4)
620     WRITE(TTO,8080)

        DO 640 I=4,9
            ITMP(I-3)=ISFILE(I,1)
640     CONTINUE

        CALL INPTS(IGNFN,ITMP,11,$8100,$7200)
        ILOC=LENG(IGNFN,11)+1
        IEE=777K
        CALL MAPST(IEXTN,1,3,2,IGNFN,11,ILOC,$800)
C       WRITE(TTO,660) (IGNFN(I),I=1,11)
C660    FORMAT(" QUANFILE: ",11A2)
        GOTO 900

800     WRITE(TTO,8120) IEE
        GOTO 620
```

```
C********************************************VVVVV
900     CALL OPENR(IGNFN,LCH,IEE,$1000)
        GOTO 1020

1000    CALL CLOSE(LCH,IE)
        GOTO 800

C       LOOK FOR "NO  NAME"
C
1020    READ(LCH,8140,END=1000,ERR=1000) (ITXT(I),I=1,40)
        IP1=NPOS(LOOK5,4,ITXT,10,1)
        IF(IP1.LE.0) GOTO 1020

C       COUNT NUMBER OF NAMES (ANALYTES)
C
        INMS=0
1060    READ(LCH,8160,END=1000,ERR=1000) IND,(ITXT(I),I=1,35)
        IF(IND.LE.0) GOTO 1100
        NIND=IND
        INMS=INMS+1
C       IF(NIND.NE.INMS) GOTO 1000
        GOTO 1060

C       LOOK FOR "NO  M/E"
C
1100    READ(LCH,8140,END=1000,ERR=1000) (ITXT(I),I=1,40)
        IP1=NPOS(LOOK6,4,ITXT,10,1)
        IF(IP1.LE.0) GOTO 1100

C       WRITE(TTO) " NAMES COUNTED.  NOW READING REFERENCES."

        IREF=0
        INM=0
        ISEG=0
1200    READ(LCH,8180,END=1000,ERR=1000) IND,(ITMP(I),I=1,9),
     1      IREF,(ITXT(I),I=1,25)
        INM=INM+1
C       LOOK FOR "NOT FOUND"
        IP1=NPOS(LOOK7,5,ITMP,9,1)
        IF(IP1.LE.0) GOTO 1280

C       REPAIR REFERENCE ERRORS FOR "NOT FOUND" COMPOUNDS.
C       THIS PRESUMES THAT ANALYTES  FOLLOW THEIR PARTICULAR
C       INT.STD.
C
        WRITE(TTO) " <7>REFERENCE ERROR REPAIRED"
        IF(INM.GT.1) IARRAY(INM)=IARRAY(INM-1)
        IF(INM.EQ.1) IARRAY(INM)=1
        GOTO 1300

C       SET INTERNAL STANDARD REFERENCE
C
1280    IARRAY(INM)=IREF

1300    IF(INM.NE.IND) ISEG=ISEG+1
        IF(INM.LT.INMS) GOTO 1200

        CALL POSITION(LCH,1,-1,-1,IEE,$1320)
1320    CALL CLOSE(LCH,IE)
C*******************************************^^^^^
```

```
        IF(ISEQ.EQ.0) GOTO 1330
        WRITE(TTO) " <7>SEQUENCE MISMATCH ENCOUNTERED IN QUAN FILE."
        GOTO 620

1330    WRITE(TTO,8190) INMS

        RMDL=5.0
        XMDL=5.0
        A=5.0
        B=500.0
        C=0.1
        D=5.0
        RMEAN=0.0
        SIGR=2.0
        XLSA2=20.0
        SMEAN=75.0
        SIGS=15.0
        XLCT=20.0
        CMEAN=75.0
        SIGC=15.0

        ITMP(1)="UG"
        ITMP(2)="<57>L"
        ITMP(3)=0

C       DEFINE UNITS
C
1340    CALL INPTS(KUNITS,ITMP,3,$8230,$7200)
        CALL PADST(KUNITS,3)

C       DEFINE RMDL
C
        XTMP=RMDL
        CALL INPTF(RMDL,XTMP,$8240,$7200)

C       DEFINE XMDL
C
        XTMP=XMDL
        CALL INPTF(XMDL,XTMP,$8250,$7200)

C       DEFINE DUPLICATE MINIMUM
C
        XTMP=A
        CALL INPTF(A,XTMP,$8260,$7200)

C       DEFINE DUPLICATE MAXIMUM
C
        XTMP=B
        CALL INPTF(B,XTMP,$8270,$7200)

C       DEFINE DUPLICATE SLOPE
C
        XTMP=C
        CALL INPTF(C,XTMP,$8280,$7200)

C       DEFINE DUPLICATE INTERCEPT
C
```

```
        XTMP=D
        CALL INPTF(D,XTMP,$8290,$7200)
C       DEFINE REAGENT MEAN
C
        XTMP=RMEAN
        CALL INPTF(RMEAN,XTMP,$8320,$7200)
C       DEFINE REAGENT SIGMA
C
        XTMP=SIGR
        CALL INPTF(SIGR,XTMP,$8340,$7200)
C       DEFINE REAGENT SIGMA
C
        XTMP=SIGR
        CALL INPTF(SIGR,XTMP,$8340,$7200)
C       DEFINE SPIKE AMOUNT
C
        XTMP=XLSA2
        CALL INPTF(XLSA2,XTMP,$8360,$7200)
C       DEFINE SPIKE MEAN RECOVERY AMOUNT
C
        XTMP=SMEAN
        CALL INPTF(SMEAN,XTMP,$8380,$7200)
C       DEFINE SPIKE SIGMA RECOVERY
C
        XTMP=SIGS
        CALL INPTF(SIGS,XTMP,$8400,$7200)
C       DEFINE CONTROL STD AMOUNT
C
        XTMP=XLCT
        CALL INPTF(XLCT,XTMP,$8420,$7200)
C       DEFINE CONTROL STD MEAN RECOVERY
C
        XTMP=CMEAN
        CALL INPTF(CMEAN,XTMP,$8440,$7200)
C       DEFINE CONTROL STD SIGMA RECOVERY
C
        XTMP=SIGC
        CALL INPTF(SIGC,XTMP,$8460,$7200)
        GOTO 1700

1640    WRITE(TTO) " <7>"
1700    IDEF="N<0>"
        CALL INPTS(IANS,IDEF,1,$8480,$7200)
        IF(IANS.EQ."Y<0>") GOTO 1340
        IF(IANS.NE."N<0>") GOTO 1640

        WRITE(TTO,8500) (IGCFN(I),I=1,8)
```

```
C*********************************************VVVVV
C       OPEN QC/MT FOR FIRST TIME
C
        CALL MAPST(IEXTC, 1, 3, 2, IGCFN, 8, 10, $440)
        CALL OPENW(IGCFN, ICH, IEE, $7000)

        WRITE(ICH, 8520) (IGCNM(I), I=1, 4), (KUNITS(I1), I1=1, 3), INMS, NHOFF,
     1      NRECS, (IPRDT(J), J=1, 4), (INIT(J1), J1=1, 3), (IPRDT(K), K=1, 4),
     1      (INIT(K1), K1=1, 3)

        CALL OPENR(IGNFN, LCH, IEE, $1800)
        GOTO 2000

1800    CALL RESET
        WRITE(TTO, 8120) IEE
        GOTO 7000

C       FIND "NO  NAME"
C
2000    READ(LCH, 8140, END=1800, ERR=1800) (ITXT(I), I=1, 40)
        IP1=NPOS(LOOK5, 4, ITXT, 10, 1)
        IF(IP1.LE.0) GOTO 2000

C       READ NAME AND NUMBER
C
        INM=0
2040    READ(LCH, 8160, END=1800, ERR=1800) IND, (ITXT(I), I=1, 35)
        INM=INM+1

C       WRITE NAME AND NUMBER
C
        WRITE(ICH, 8540) IND, (ITXT(I), I=1, 35)

C       CHECK INT.STD REFERENCE AND SET IUSE=3 IF FOUND
C
        IUSE=0
        IF(IARRAY(IND).EQ.IND) IUSE=3           ; INT STD

        Y1=RMDL+D3
        Y2=XMDL+D3
        Y3=A+D4
        Y4=B+D4
        Y5=C+D4
        Y6=D+D4
        WRITE(ICH, 8560) (ICAS(I), I=1, 6), IUSE, Y1, Y2, Y3, Y4, Y5, Y6, KEYD

        X1=RMEAN+D3
        X2=SIGR+D3
        X3=XLSA2+D3
        X4=SMEAN+D2
        X5=SIGS+D2
        X6=XLCT+D3
        X7=CMEAN+D2
        X8=SIGC+D2
C                         REAGENT  *  SPIKE      * CONTRL STD
        WRITE(ICH, 8580) X1, X2, KEYD, X3, X4, X5, KEYD, X6, X7, X8, KEYD

        IF(INM.LT.INMS) GOTO 2040

        CALL POSITION(LCH, 1, -1, -1, IEE, $2800)
2800    CALL CLOSE(LCH, IE)
        CALL POSITION(ICH, 1, -1, -1, IEE, $2820)
2820    CALL CLOSE(ICH, IE)
C*********************************************^^^^^
```

```
2900    CALL LOADIT("DMPG2")

7000    WRITE(TTO,8600) IEE
        WRITE(TTO) " <7>FATAL ERROR!"
7200    CALL RESET
        CALL LOADIT("EXEC")

8000    FORMAT("1",/," GC<57>METHOD TEMPLATE EDITOR MODULE (DMPG)",/)
8020    FORMAT("  YOUR INITIALS: [",3A2,"] > ",Z)
8040    FORMAT("  GC<57>METHOD TEMPLATE NAME [",4A2,"] > ",Z)
8060    FORMAT(" <7>GC<57>MT ACCESS ERROR: ",OI6)
8080    FORMAT(/," YOU ARE CREATING A NEW GC<57>METHOD TEMPLATE.",/,
     1       " PLEASE IDENTIFY A RENAMED QUAN.99 FILE REPRESENTING",/,
     1       " A FIXED LIST OF STANDARDS. ALSO ACCEPT OR MODIFY",/,
     1       " THE DEFAULT QC LIMITS FOR THE GC<57>MT.",/)
8100    FORMAT("  QUAN REPORT OR SAMPLE NAME [",11A2,"] > ",Z)
8120    FORMAT(/," <7>UNABLE TO PROCESS QUAN FILE. ERROR: ",OI6)
8140    FORMAT(40A2)
8160    FORMAT(I3,2X,35A2)
8180    FORMAT(I3,9A2,I4,25A2)
8190    FORMAT(" QUAN FILE HAS",I4," ENTRIES.")
8230    FORMAT("    UNITS OF MEASURE: [",3A2,"] > ",Z)
8240    FORMAT(" * REAGENT MDL: [",F7.2,"] > ",Z)
8250    FORMAT(" * MATRIX MDL:  [",F7.2,"] > ",Z)
8260    FORMAT(" * RANGE MINIMUM:       [",F10.3,"] > ",Z)
8270    FORMAT(" * RANGE MAXIMUM:       [",F10.3,"] > ",Z)
8280    FORMAT(" * DUPLICATE SLOPE:     [",F10.3,"] > ",Z)
8290    FORMAT(" * DUPLICATE INTRCPT:   [",F10.3,"] > ",Z)
8320    FORMAT(" * MEAN REAGENT BLNK:   [",F7.2,"] > ",Z)
8340    FORMAT(" * REAGENT STD.DEV:     [",F7.2,"] > ",Z)
8360    FORMAT(" * SPIKE CONC. ADDED:   [",F8.2,"] > ",Z)
8380    FORMAT("    MEAN SPIKE RECOV:   [",F6.1,"%] > ",Z)
8400    FORMAT("    SPIKE STD.DEV:      [",F6.1,"%] > ",Z)
8420    FORMAT(" * CONTROL STD.CONC:    [",F8.2,"] > ",Z)
8440    FORMAT("    MEAN CNT.STD RECOV:  [",F6.1,"%] > ",Z)
8460    FORMAT("    CNT.STD STD.DEV:     [",F6.1,"%] > ",Z)
8480    FORMAT("  DID YOU MAKE A MISTAKE DEFINING DEFAULTS [",A2,"] > ",Z)
8500    FORMAT("1"," THE SKELETON GC<57>MT: ",8A2," WILL NOW BE CREATED.",
     1      /,"  PLEASE USE EDIT OPTIONS TO COMPLETE THE DESCRIPTIONS.",/)
8520    FORMAT(" ",4A2,3A2,3I4,1X,4A2,1X,3A2,1X,4A2,1X,3A2,"*")
8540    FORMAT(" ",I3,1X,35A2,"+")
8560    FORMAT(" ",6A2,"#",I3,"!",2(F7.2),"?",4(F10.3),A1)
8580    FORMAT(" ",2F7.2,A1,2(F8.2,2F6.1,A1))
8600    FORMAT(" <7>GC<57>MT ERROR: ",OI6)

        END
```

```
C       DMPG2.FR  GC/METHOD TEMPLATE EDITOR - ROUTER AND REPORTER
C       D.RYAN 3/82    LAST REVISION: 09/23/82

        PARAMETER TTO=10,LPT=2,DSK=1,ICH=3,LCH=4,MAXM=120,MAXSM=25

        COMMON /CJNK1/
     1  NAME(36),ICAS(6),ITMP(40),ITXT(70),IGNFN(11),KUNITS(3),
     1  ICINT(3),ICDT(4),IUINT(3),IUDT(4),IARRAY(MAXM),
     1  IYR,IMO,IDAY,IX,KEYD,NHOFF,NRECS,IANS,D1,D2,D3,D4,ILEN,I; IEE,
     1  ILOC,IP1,INMS,NIND,IND,IREF,INM,ISEG,RMDL,XMDL,A,B,C,D,
     1  RMEAN,SIGR,XLSA2,SMEAN,SIGS,XLCT,CMEAN,SIGC,XTMP,IDEF,
     1  IUSE,X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,Y1,Y2,Y3,Y4,NM,NO,NR,
     1  MOP,ICHN,IOPT,ISBFLD,INSTR,INSTP,ISTRT,ISTOP,NREC,IBIT,
     1  KEYR,KEYS,KEYC,KCHAN,IE,I1,I2

        COMMON /CJNK2/
     1  LOOK5(4),LOOK6(4),LOOK7(5),IEXTC(2),IEXTN(2),IUPT(32),
     1  IGST3(3),NUMBS(5)

        COMMON /CJNK3/
     1  IGCDSC(11),IGCFN(8),INIT(3),IH1,IM1,IS1

        COMMON /CJNK4/
     1  IGCNM(4),IPRDT(4)

        COMMON /CMAIN/ NPROG,IDFSW,IDFSW1,ITEMP(15),
     1      IDFILE(12,6),ISFILE(11,4),INSTD(3),IREFTABLE(11),
     1      IADD1,IADD2,ISUB1,ISUB2,ISUB3,ISUB4,JADD1,
     1      ILIBNAME,ILIBNUMB,IBUF,ISCN(45),IACG(100),IDAT(45)

        DATA IUPT/"NOT SPECIFIED   NORMAL SPIKE     ",
     1      "SURROGATE SPIKE INTERNAL STND.   "/
        DATA IGST3/"MRGSXH"/
        DATA NUMBS/"012345678<0>"/

        WRITE(TTO,100) IGCDSC(1),(IGCDSC(I),I=4,9),IGCDSC(10)
100     FORMAT(" FILE: ",A2,":",6A2,".",A2)
        GOTO 700

300     CALL RESET
400     WRITE(TTO,8000) IEE
        PAUSE ERROR EXIT! CR WHEN READY.
        GOTO 1700

700     CONTINUE
        WRITE(TTO,8010) IH1,IM1,IS1
        CALL GETTI(IYR,IMO,IDAY,IH2,IM2,IS2,$720,IX,IX,IX,IX,IX,IX)
        IPRDT(1)=(IMO/10+48)*256+MOD(IMO,10)+48
        IPRDT(2)=47*256+IDAY/10+48
        IPRDT(3)=(MOD(IDAY,10)+48)*256+47
        IPRDT(4)=(MOD(IYR/10,10)+48)*256+MOD(IYR,10)+48
        WRITE(TTO,8010) IH2,IM2,IS2
        GOTO 800

720     WRITE(TTO) " <7>PLEASE SET TIME AT MSDS"
        CALL LOADIT("EXEC")

800     WRITE(TTO,8020)
C******************************************VVVVV
        CALL OPENR(IGCFN,ICH,IEE,$300)
        CALL POSITION(ICH,0,0,0,IEE,$300)
        READ(ICH,8040,END=300,ERR=300) IGCNM,KUNITS,NM,NO,NR,ICDT,ICINT,
     1      IUDT,IUINT
        CALL POSITION(ICH,1,-1,-1,IER,$820)
820     CALL CLOSE(ICH,IE)
C*******************************************^^^^^
        ICHN=TTO
        CALL GOSUB($7000)
        GOTO 980
```

```
960     WRITE(TTO) " <7>"
        WRITE(TTO,8060)
980     MOP=0
        IANS="H<0>"
        CALL INPTS(ITXT,IANS,1,$8080,$1900)
        MOP=NPOS(IANS,1,IGST3,3,1)
        IF(MOP.LE.0.OR.MOP.GT.5) GOTO 960

        NRECS=NR
        NHOFF=NO
        D1=0.5
        D2=0.05
        D3=0.005
        D4=0.0005

C       M=MODIFY TEMPLATE
C       R=REPORT TEMPLATE CONTENTS
C       G=GET ANOTHER TEMPLATE NAME (FROM DMPG1)
C       S=ASSIMILATE SFC LIMITS
C       X=EXIT TO DMPC

C                M     R     G     S     X     H
        GOTO (2000,4000,1000,1500,1900,960) MOP
        GOTO 960

1000    CALL LOADIT("DMPG1")
        GOTO 1700

1500    CALL LOADIT("DMPG3")

1700    IER=777
        WRITE(TTO,1800) IER
1800    FORMAT(" <7>FATAL ERROR ",O14)
1900    CALL RESET
        CALL LOADIT("DMPC")
        STOP

C       MODIFY CONTENTS OF GC METHOD TEMPLATE
C
2000    ICHN=TTO
        WRITE(TTO,8100)
        IOPT=0K
2040    IANS="0<0>"
        CALL INPTS(IX,IANS,1,$8120,$800)
        IF((IANS.AND.377K).GT.0) GOTO 2060
        ISBFLD=NPOS(IANS,1,NUMBS,5,1)-1
        IF(ISBFLD.EQ.0.AND.IOPT.GT.0) GOTO 2200
        IF(ISBFLD.GE.1.AND.ISBFLD.LE.9) GOTO 2120
2060    WRITE(TTO) " <7>"
        GOTO 2040
2120    IOPT=(IOPT.OR.2**(ISBFLD-1))
        GOTO 2040

2200    WRITE(TTO,8140) NM
        INSTR=1
        INSTP=NM
        CALL INPTI(ISTRT,INSTR,$8160,$800)
        CALL INPTI(ISTOP,INSTP,$8180,$800)
C       ISTRT=2
C       ISTOP=5
        IF(ISTRT.GE.1.AND.ISTOP.LE.NM.AND.ISTRT.LE.ISTOP) GOTO 2300
        WRITE(TTO) " <7>"
        GOTO 2200

2300    INM=ISTRT

2320    IF(INM.GT.ISTOP) GOTO 3660
        WRITE(TTO,8020)
C       NREC=TRUE RECORD NUMBER MINUS 1
        NREC=INM-1
C       NRECS=SIZE OF THREE LINE RECORD (INCLUDING CR'S)
        NRECS=NR
C       NHOFF=THE HEADER OFFSET LENGTH
        NHOFF=NO
```

```
C**********************************************VVVVV
        CALL OPENR(IGCFN,ICH,IER,$3600)
        CALL POSITION(ICH,NREC,NRECS,NHOFF,IER,$3600)
        READ(ICH,8200,END=3600,ERR=3600) IND,(NAME(I),I=1,35)
        READ(ICH,8220,END=3600,ERR=3600) ICAS,IUSE,RMDL,
     1      XMDL,A,B,C,D,KEYD
        READ(ICH,8240,END=3600,ERR=3600) RMEAN,SIGR,KEYR,XLSA2,SMEAN,
     1      SIGS,KEYS,XLCT,CMEAN,SIGC,KEYC
        CALL POSITION(ICH,1,-1,-1,IEE,$2360)
2360    CALL CLOSE(ICH,IE)
C********************************************^^^^^

2400    CALL GOSUB($7400)

        IBIT=1K
        ISBFLD=1
2420    IF((IBIT.AND.IOPT).EQ.0) GOTO 3300
C             1    2    3    4    5    6    7    8
        GOTO (2500,2600,2700,2800,2900,3000,3100,3200) ISBFLD

C       COMPOUND NAME MODIFICATION
C
2500    CALL INPTS(ITXT,NAME,35,$8260,$3640)
        CALL PADST(NAME,35)
        GOTO 3300

C       COMPOUND CAS # MODIFICATION
C
2600    CALL INPTS(ITXT,ICAS,6,$8280,$3640)
        CALL NCAS(ICAS,6,$2640)
        CALL PADST(ICAS,6)
        GOTO 3300
2640    WRITE(TTO) " <7>"
        GOTO 2600

C       USE CODE MODIFICATION
C
2700    CALL INPTI(IX,IUSE,$8300,$3640)
        IF(IUSE.GE.0.AND.IUSE.LE.3) GOTO 3300
        WRITE(TTO) " <7>"
        IUSE=0
        GOTO 2700

C       MDL MODIFICATION
C
2800    CALL INPTF(X,RMDL,$8320,$3640)
        CALL INPTF(X,XMDL,$8340,$3640)
        IF(RMDL.GE.0.0.AND.XMDL.GE.0.0) GOTO 3300
        WRITE(TTO) " <7>"
        GOTO 2800

C       DUPLICATE LIMIT MODIFICATION
C
2900    CALL INPTF(X,A,$8360,$3640)
        CALL INPTF(X,B,$8370,$3640)
        CALL INPTF(X,C,$8380,$3640)
        CALL INPTF(X,D,$8390,$3640)
        KEYD="A<0>"
        IF((A.GT.0.0).AND.(B.GT.A)) GOTO 3300
        WRITE(TTO) " <7>"
        GOTO 2900

C       REAGENT BLANK LIMIT
C
3000    CALL INPTF(X,RMEAN,$8400,$3640)
        CALL INPTF(X,SIGR,$8420,$3640)
        KEYR="A<0>"
        IF(RMEAN.GE.0.0.AND.SIGR.GE.0.0) GOTO 3300
        WRITE(TTO) " <7>"
        GOTO 3000
```

```
C       SPIKE SAMPLE LIMITS
C
3100    CALL INPTF(X, XLSA2, $8440, $3640)
        IF((IUSE. EQ. 2). AND. (XLSA2. LT. XMDL)) GOTO 3120
        CALL INPTF(X, SMEAN, $8460, $3640)
        CALL INPTF(X, SIGS, $8480, $3640)
        KEYS="A<0>"
        IF(XLSA2. GE. 0. 0. AND. SMEAN. GE. 0. 0. AND. SIGS. GE. 0. 0) GOTO 3300
3120    WRITE(TTO) " <7>"
        GOTO 3100

C       CONTROL STANDARD LIMITS
C
3200    CALL INPTF(X, XLCT, $8500, $3640)
        CALL INPTF(X, CMEAN, $8460, $3640)
        CALL INPTF(X, SIGC, $8480, $3640)
        KEYC="A<0>"
        IF(XLCT. GE. 0. 0. AND. CMEAN. GE. 0. 0. AND. SIGC. GE. 0. 0) GOTO 3300
        WRITE(TTO) " <7>"
        GOTO 3200

C       NEXT SUBFIELD
C
3300    ISBFLD=ISBFLD+1
        IF(ISBFLD. GT. 8) GOTO 3400
        IBIT=ISHFT(IBIT, 1)
        GOTO 2420

3400    WRITE(TTO, 8020)
        CALL GOSUB($7400)

3420    IANS="N<0>"
        CALL INPTS(IX, IANS, 1, $8520, $3640)
        IF(IANS. EQ. "N<0>") GOTO 3440
        IF(IANS. NE. "Y<0>") GOTO 3430
        WRITE(TTO, 8020)
        GOTO 2400
3430    WRITE(TTO) " <7>"
        GOTO 3420

C*******************************************VVVVV
3440    CALL OPEN(IQCFN, ICH, IER, $3600)
        CALL POSITION(ICH, NREC, NRECS, NHOFF, IER, $3600)
        X1=RMDL+D3
        X2=XMDL+D3
        Y1=A+D4
        Y2=B+D4
        Y3=C+D4
        Y4=D+D4
        X3=RMEAN+D3
        X4=SIGR+D3
        X5=XLSA2+D3
        X6=SMEAN+D2
        X7=SIGS+D2
        X8=XLCT+D3
        X9=CMEAN+D2
        X10=SIGC+D2
        WRITE(ICH, 8540) IND, (NAME(I), I=1, 35)
        WRITE(ICH, 8560) (ICAS(I), I=1, 6), IUSE, X1, X2, Y1, Y2, Y3, Y4, KEYD
        WRITE(ICH, 8580) X3, X4, KEYR, X5, X6, X7, KEYS, X8, X9, X10, KEYC
        CALL POSITION(ICH, 1, -1, -1, IER, $3500)
3500    CALL CLOSE(ICH, IE)
C*******************************************^^^^^

        INM=INM+1
        GOTO 2320

C       ERROR PROCESSING
C
3600    CALL RESET
        WRITE(TTO, 8600) IER
        GOTO 800

C       ^D PROCESSING
C
3640    WRITE(TTO, 8620)

3660    IF(INM. LE. ISTRT) GOTO 800
```

```
C*******************************************VVVVV
C       UPDATE HEADER IF SOMETHING CHANGED
C
        CALL OPEN(IGCFN, ICH, IER, $3600)
        CALL POSITION(ICH, 0, 0, 0, IEE, $400)
        WRITE(ICH, 8640) (IGCNM(I), I=1, 4), (KUNITS(I), I=1, 3), NM, NO, NR,
     1      (ICDT(I), I=1, 4), (ICINT(I), I=1, 3), (IPRDT(I), I=1, 4),
     1      (INIT(I), I=1, 3)
        CALL POSITION(ICH, 1, -1, -1, IEE, $3720)
3720    CALL CLOSE(ICH, IE)
C*******************************************^^^^^

        GOTO 800

C*******************************************VVVVV
C       REPORT CONTENTS OF GC METHOD TEMPLATE
C
4000    CALL OPENR(IGCFN, ICH, IEE, $4200)
        CALL POSITION(ICH, 0, 0, NHOFF, IEE, $4200)
C       READ(ICH, 8040, END=4200, ERR=4200) IGCNM, KUNITS, NM, NO, NR, ICDT,
C     1      ICINT, IUDT, IUINT

        CALL DELETE("1:DMPRPTG.99", IEE, $4020)
4020    CALL OPENW("1:DMPRPTG.99", LCH, IEE, $4400)
        ICHN=LCH
        CALL GOSUB($7000)
        INM=0

        WRITE(TTO) " GENERATING REPORT"
4040    INM=INM+1
        IF(INM.GT.NM) GOTO 4100

        READ(ICH, 8200, END=4200, ERR=4200) IND, (NAME(I), I=1, 35)
        READ(ICH, 8220, END=4200, ERR=4200) (ICAS(I), I=1, 6), IUSE, RMDL, XMDL,
     1      A, B, C, D, KEYD
        READ(ICH, 8240, END=4200, ERR=4200) RMEAN, SIGR, KEYR, XLSA2, SMEAN, SIGS,
     1      KEYS, XLCT, CMEAN, SIGC, KEYC
        IF(MOD(INM, 5).NE.0) GOTO 4080
        WRITE(ICHN, 8020)
4080    CALL GOSUB($7400)
        GOTO 4040

4100    WRITE(ICHN, 8020)
        CALL CLOSE(LCH, IE)
        CALL POSITION(ICH, -1, -1, 0, IEE, $4120)
4120    CALL CLOSE(ICH, IE)
C*******************************************^^^^^

        WRITE(TTO) " PRINTING REPORT"
        KCHAN=LPT
        CALL OPLPT(KCHAN, $4180)
        CALL OPENR("1:DMPRPTG.99", DSK, IER, $4160)
        CALL FCOPY(DSK, KCHAN, $4160, IE, IER)
4160    CALL CLOSE(DSK, IE)
        CALL CLOSE(KCHAN, IE)
        GOTO 800

4180    WRITE(TTO) " <7>REPORT IN 1:DMPRPTG.99"
        GOTO 800

4200    CALL CLOSE(ICH, IE)
        WRITE(TTO, 8000) IEE

4400    CALL RESET
        WRITE(TTO) " <7>UNABLE TO GENERATE REPORT!"
C       CALL TIME(5000)
        PAUSE CR TO CONTINUE
        GOTO 800

C
C       REPORT HEADER WRITER
C
7000    WRITE(ICHN, 8680) (IPRDT(I), I=1, 4)
        WRITE(ICHN, 8700) (IGCNM(I), I=1, 4), (KUNITS(I), I=1, 3)
        WRITE(ICHN, 8720) (ICDT(I), I=1, 4), (IUDT(I), I=1, 4)
        WRITE(ICHN, 8740) (ICINT(I), I=1, 3), (IUINT(I), I=1, 3)
        WRITE(ICHN, 8760) NM
        CALL GORET
```

```
7400    WRITE(ICHN,8780) INM
        WRITE(ICHN,8800) (NAME(I),I=1,35)
        I1=IUSE*8+1
        I2=(IUSE+1)*8
        X1=RMDL+D3
        X2=XMDL+D3
        Y1=A+D4
        Y2=B+D4
        Y3=C+D4
        Y4=D+D4
        X3=RMEAN+D3
        X4=SIGR+D3
        X5=XLSA2+D3
        X6=SMEAN+D2
        X7=SIGS+D2
        X8=XLCT+D3
        X9=CMEAN+D2
        X10=SIGC+D2
        WRITE(ICHN,8820) (ICAS(I),I=1,6),IUSE,(IUPT(I),I=I1,I2)
        WRITE(ICHN,8840) KEYD
        WRITE(ICHN,8860) X1,Y1,Y2
        WRITE(ICHN,8880) X2,Y3,Y4
        WRITE(ICHN,8900)
        WRITE(ICHN,8920) KEYR,X3,X4
        WRITE(ICHN,8940) KEYS,X5,X6,X7
        WRITE(ICHN,8960) KEYC,X8,X9,X10
        CALL GORET

8000    FORMAT(" <7>GC<57>MT ACCESS ERROR:  ",OI4," PUNT!")
8010    FORMAT(" ",I2,":",I2,":",I2)
8020    FORMAT("1")
8040    FORMAT(4A2,3A2,3I4,1X,4A2,1X,3A2,1X,4A2,1X,3A2,2X)
8060    FORMAT(//," GC<57>MT EDIT OPTIONS:",/,
     1      "  M -MODIFY CURRENT GC<57>MT FIELDS(S)",/,
     1      "  R -REPORT CONTENTS OF GC<57>MT",/,
     1      "  S -UPDATE GC<57>MT WITH SFC SUPPLIED LIMITS",/,
     1      "  G -GET ANOTHER GC<57>MT TO PROCESS",/,
     1      "  X -EXIT TO DMP COMMAND LEVEL",/,
     1      " ^D -EXITS TO MSDS",//)
8080    FORMAT(" OPTION (MRSGXH) [",A2,"] > ",Z)
8100    FORMAT("1",//," EDIT SUBFIELD INDEX:",/,
     1      "  0 -DONE SELECTING SUBFIELDS",/
     1      "  1 -COMPOUND NAME",/,
     1      "  2 -CAS NUMBER",/,
     1      "  3 -USE CODE",/,
     1      "  4 -MDL",/,
     1      "  5 -DUPLICATE PARAMETERS",/,
     1      "  6 -REAGENT BLANK PARAMETERS",/,
     1      "  7 -SPIKE PARAMETERS",/,
     1      "  8 -CONTROL STANDARD PARAMETERS",/,
     1      " TO MODIFY THE PARAMETERS IN A GIVEN",/,
     1      " SUBFIELD, JUST ENTER THE SUBFIELD ",/,
     1      " NUMBER WHEN PROMPTED. MORE THAN ONE",/,
     1      " SUBFIELD MAY BE SELECTED. ",/,
     1      " ^D RETURNS TO EDIT OPTIONS. ",//)
8120    FORMAT(" SUBFIELD # [",A2,"] > ",Z)
8140    FORMAT(//," THERE ARE ",I4," COMPOUNDS IN THE GC<57>MT",/,
     1      " PLEASE ENTER THE RANGE YOU WISH TO MODIFY:")
8160    FORMAT(" STARTING COMPOUND [",I4,"] > ",Z)
8180    FORMAT(" STOPPING COMPOUND [",I4,"] > ",Z)
8200    FORMAT(I3,1X,35A2,2X)
8220    FORMAT(6A2,1X,I3,1X,2F7.2,1X,4F10.3,A1,1X)
8240    FORMAT(2F7.2,A1,F8.2,2F6.1,A1,F8.2,2F6.1,A1,1X)
8260    FORMAT(" COMPOUND NAME:",/," [",35A2,"]",/," >",Z)
8280    FORMAT(" CAS #:  [",6A2,"] > ",Z)
8300    FORMAT(" USE CODE:",/,
     1          "  0 -NONE",/,
     1          "  1 -SPIKE AND<57>OR CONTROL",/,
     1          "  2 -SURROGATE SPIKE AND<57>OR CONTROL",/,
     1          "  3 -INTERNAL STANDARD",/,
     1          " USE CODE:    [",I4,"] > ",Z)
8320    FORMAT(" METHOD DETECTION LIMITS:",/,
     1          "  REAGENT MDL [",F7.2,"] > ",Z)
8340    FORMAT("  MATRIX MDL  [",F7.2,"] > ",Z)
```

```
8360    FORMAT(" DUPLICATE LIMITS:",/,
     1          "  RANGE MIN  (A) [",F10.3,"] > ",Z)
8370    FORMAT("  RANGE MAX  (B) [",F10.3,"] > ",Z)
8380    FORMAT("  SLOPE      (C) [",F10.3,"] > ",Z)
8390    FORMAT("  INTERCEPT  (D) [",F10.3,"] > ",Z)
8400    FORMAT(" REAGENT LIMITS:",/,
     1          "  MEAN CONC  [",F7.2,"] > ",Z)
8420    FORMAT("  STD.DEV.    [",F7.2,"] > ",Z)
8440    FORMAT(" SPIKE PARAMS:",/,
     1          "  SPIKE CONC [",F8.2,"] > ",Z)
8460    FORMAT("  MEAN %RECV [",F6.1,"%] > ",Z)
8480    FORMAT("  STD.DEV %  [",F6.1,"%] > ",Z)
8500    FORMAT(" CONTROL STAND. PARAMS:",/,
     1          "  STD.CONC   [",F8.2,"] > ",Z)
8520    FORMAT(" MISTAKES ON DISPLAYED PARAMS [",A2,"] > ",Z)
8540    FORMAT(" ",I3,1X,35A2,"+")
8560    FORMAT(" ",6A2,"#",I3,"!",2F7.2,"?",4F10.3,A1)
8580    FORMAT(" ",2F7.2,A1,F8.2,2F6.1,A1,F8.2,2F6.1,A1)
8600    FORMAT(" <7>FILE I<57>O ERROR: ",OI6)
8620    FORMAT(" <7>TASK TERMINATING ABRUPTLY!")
8640    FORMAT(" ",4A2,3A2,3I4,1X,4A2,1X,3A2,1X,4A2,1X,3A2,"*")
C8660   FORMAT(4A2,3A2,3I4,1X,4A2,1X,3A2,1X,4A2,1X,3A2,2X)
8680    FORMAT(//,T26,"QC<57>METHOD TEMPLATE",/,T31,4A2,//)
8700    FORMAT(" ",T6,"NAME: ",4A2,T42,"DEFAULT UNITS (U): ",3A2)
8720    FORMAT(" ",T6,"DATE CREATED: ",4A2,T42,"LAST UPDATE: ",4A2)
8740    FORMAT(" ",T6,"CREATED BY: ",3A2,T42,"CHANGED BY: ",3A2)
8760    FORMAT(" ",T6,"NO.OF COMPOUNDS: ",I4,T42,"EDIT SUBFIELDS: (N)",//)
8780    FORMAT(" COMPOUND NUMBER: ",I4)
8800    FORMAT(" (1) ",35A2)
8820    FORMAT(" (2) CAS #: ",6A2,T34," (3) USE CODE: ",I2,"=",8A2)
8840    FORMAT(" (4) METHOD DETECTION LIMITS:",T34,
     1          " (5) DUPLICATE LIMITS <57>",A1,"<57>:")
8860    FORMAT("     REAGENT MDL: ",F7.2,T34," MIN [A]: ",F10.3,
     1          "  MAX [B]: ",F10.3)
8880    FORMAT("     MATRIX MDL:  ",F7.2,T34," SLP [C]: ",F10.3,
     1          "  INT [D]: ",F10.3)
8900    FORMAT(" ",/,T32,"EXPECTED        MEAN      STD.DEV")
8920    FORMAT(" (6) REAGENT BLANK LIMITS:  <57>",A1,"<57>",T45,F7.2,
     1          " U    ",F7.2," U")
8940    FORMAT(" (7) SPIKED SAMPLE LIMITS:  <57>",A1,"<57>",T30,F8.2,
     1          " U    ",F6.1," %      ",F6.1," %")
8960    FORMAT(" (8) CNTRL STANDARD LIMITS: <57>",A1,"<57>",T30,F8.2,
     1          " U    ",F6.1," %      ",F6.1," %",/)

        END
```

```
C       DMPR1.FR  REPLICATE ANALYSIS MODULE - NAMELIST ENTRY/VARIFICATION
C       D.RYAN 2/82     LAST REVISION: 09/23/82

        PARAMETER TTO=10,LPT=2,MAXN=120,MAXSM=25,DSK=1,ICH=1,LCH=4,MCH=5

        COMMON /CJNK1/ INST(3),RESPF(MAXN),
     1  ISMPLE(32),IANLST(4),ICOND(32),IACCNT(4),ITMP(20),ITXT(50),
     1  IACDT(4),IACDT2(4),NAME(35),IPMTH(4),
     1  IEXDT(4),IEXDT2(4),ICKST(5),IRGBK(5),ICMNT(32),ICAS(6),NCOMP(2),
     1  IANS(5),IYR,IDAY,IMO,IX,MODEP,LMD,LN,LNN,
     1  NM,NO,NN,IC,ICHR,IPOS,ILEN,NIND,ICODE,X,ITP,
     1  LOOP,MI,IP,CN,AR,IDT,IRRT,F,ICD,M1,M2,I1,I2,INDG,ICHAR,
     1  IND,IUSE,RMDL,XMDL,A,B,C,D,KEYD,RMEAN,SIGR,KEYR,XLSA2,
     1  SMEAN,SIGS,KEYS,XLCT,CMEAN,SIGC,KEYC,A1,A2,A3,XGC,SGC,YMDL

        COMMON /CJNK2/ IEXTC(2),IEXTR(2),IGST1(5),IGST2(3),
     1  T(24),ITYPS(32),KH1(36),KH2(36),KH3(36),KH1A(11),KH2A(11),
     1  KH3A(11),KH1B1(6),KH1B2(6),KH2B(6),KH3B(6),KH1C(3),KH2C(3),
     1  KH3C(3),ITEMS(64)

        COMMON /CJNK3/
     1  IGCDSC(11),IGCFN(10),IFLDSC(11),IFLNM(8),IFLFN(10),INLDSC(11),
     1  INLNM(8),AMAT(MAXN,3),KUNITS(3),IS(11,MAXSM),ITYP,MODE,M,NI,NG,
     1  NX,NO,IH1,IM1,IS1,N,FMDL

        COMMON /CJNK4/
     1  IGCNM(4),IPRDT(4)

        COMMON /CMAIN/ NPROG,IDFSW,IDFSW1,ITEMP(15),
     1  IDFILE(12,6),ISFILE(11,4),INSTD(3),IREFTABLE(11),
     1  IADD1,IADD2,ISUB1,ISUB2,ISUB3,ISUB4,JADD1,
     1  LIBNAME,LIBNUMB,IBUF,ISCN(45),IACG(100),IDAT(45)

        DATA IEXTC/".GC<0>"/
        DATA IEXTR/".RD<0>"/
        DATA IGST1/"UNCFMTRA<0>H"/
        DATA IGST2/"KLW<0><0>H"/

        DATA T/63.657,9.925,5.841,4.604,4.032,3.707,3.499,3.355,
     1          3.250,3.169,3.106,3.005,3.012,2.977,2.947,2.921,
     1          2.898,2.878,2.861,2.845,2.831,2.819,2.807,2.797/

        DATA ITYPS/"UNKNOWN DUPED UNSPIKED USPK<53>DUP ",
     1             "CNTRL STRGNT BLKNON-SFC CAL STND"/

        DATA KH1/ 16*20040K," CAS             AVERAGE   STANDARD  % REL   "/
        DATA KH2/ " NO.  COMPOUND NAME",7*20040K,"NUMBER            VALUE ",
     1          "  DEVIATION  STDEV  "/
        DATA KH3/ 36*36475K/
        DATA KH1A/ "  KNOWN    MEAN  LIMIT "/
        DATA KH2A/ "  VALUE    %REC  RANGE "/
        DATA KH3A/ 11*36475K/
        DATA KH1B1/ "    REAGENT "/
        DATA KH1B2/ "      MATRIX "/
        DATA KH2B/ "      MDL    "/
        DATA KH3B/ 6*36475K/
        DATA KH1C/ "NO.OF "/
        DATA KH2C/ "ITEMS "/
        DATA KH3C/ "======"/
        DATA FMDL/5.0/

        DATA ITEMS/ "CONCENTRATION   CONCENTRATION   CONC,%REC,& MDLR",
     1              "REC.CONC,%REC   MATRIX MDL      REL.RET.TIME    ",
     1              "RESPONSE FACTOR RESPONSE RATIO  "/

        CALL GETTI(IYR,IMO,IDAY,IH1,IM1,IS1,*50,IX,IX,IX,IX,IX,IX)
        IPRDT(1)=(IMO/10+48)*256+MOD(IMO,10)+48
        IPRDT(2)=47*256+IDAY/10+48
        IPRDT(3)=(MOD(IDAY,10)+48)*256+47
        IPRDT(4)=(MOD(IYR/10,10)+48)*256+MOD(IYR,10)+48
        GOTO 100
```

```
50      WRITE(TTO) " <7>PLEASE SET TIME AT MSDS"
        GOTO 9500

100     WRITE(TTO, 8000)
        CALL NULST(ITMP, 1, 20)
        CALL NULST(ITXT, 1, 50)
        CALL NULST(IANS, 1, 5)
        CALL NULST(IGCDSC, 1, 11)
        CALL NULST(IGCFN, 1, 10)
        CALL NULST(IFLDSC, 1, 11)
        CALL NULST(IFLNM, 1, 8)
        CALL NULST(IFLFN, 1, 10)
        CALL NULST(INLDSC, 1, 11)
        CALL NULST(INLNM, 1, 8)

280     CALL INPTS(ITXT, IGCNM, 4, $8020, $9500)

        IGCDSC(1)="D<0>"
        IGCDSC(2)=0
        IGCDSC(3)=1
        IGCDSC(10)="GC"
        DO 400 I=1, 4
            IGCDSC(I+3)=IGCNM(I)
            IGCFN(I)=IGCNM(I)
400     CONTINUE
        CALL MAPST(IEXTC, 1, 3, 2, IGCFN, 10, 8, $500)
        CALL FSLKU(IGCDSC, $500, $500, IEE)
C       CALL MOVBT(". <0>", IGCFN, 10, 8, $500)
C       IGCFN(5)="GC"
C       WRITE(TTO, 420) (IGCFN(I), I=1, 10)
C420    FORMAT("   GCFN:  ", 10A2)
C       CALL OPENR(IGCFN, LCH, IER, $500)
C       WRITE(TTO) " GOT PAST OPEN"
        GOTO 620

500     WRITE(TTO) "    <7>GC<57>MT NOT ACCESSIBLE. "
        GOTO 280

600     WRITE(TTO, 8040)

620     IANS(1)="H<0>"
        CALL INPTS(ITXT, IANS, 5, $8080, $9500)
        MODE=NPOS(IANS, 5, IGST1, 5, 1)
        IF(MODE. EQ. 0. OR. MODE. GT. 8) GOTO 600
        MODE=MODE-1
        GOTO 900

800     WRITE(TTO, 8100)

900     IANS(1)="H<0>"
        CALL INPTS(ITXT, IANS, 5, $8140, $9500)
        LMD=NPOS(IANS, 5, IGST2, 3, 1)
        IF(LMD. EQ. 0. OR. LMD. GT. 3) GOTO 800

C=======================================================
C                NAMELIST ACCESS
C=======================================================

        NX=0
        NO=0
        IC=0
        N=0
        DO 940 J=1, MAXSM
        DO 930 I=1, 11
                IS(I, J)=0
930     CONTINUE
940     CONTINUE

C               K    L    W
        GOTO (1150, 1000, 950) LMD

C       WILDCARD NAME LIST ENTRY WHERE USER ENTERS A NAME WITH
C       A # SIGN IN IT AND THE PROGRAM SUBSTITUTES DIGITS FOR
C       THE # AND, BY SO DOING, GENERATES TEN FILE NAMES.
```

```
950     ITMP(1)="RO"
        ITMP(2)="OT"
        ITMP(3)="WD"
        ITMP(4)="#@"
        GOTO 970
960     WRITE(TTO) " <7>ERROR"
970     CALL INPTS(ITXT,ITMP,20,$8160,$9500)

C       IANS(1)="#<0>" FOR SEARCH, THEN SET TO ASCII DIGITS
        IANS(1)=21400K
        IPOS=NPOS(IANS,5,ITMP,20,1)
        IF(IPOS.LT.1.OR.IPOS.GT.12) GOTO 960

        ICHR=26400K
        DO 990 J=1,10
                ICHR=ICHR+400K              ;"0<0>" ......"9<0>"
                CALL MOVBT(ICHR,ITMP,20,IPOS,$960)
                DO 988 I=4,9
988             IS(I,J)=ITMP(I-3)
                NX=NX+1
990     CONTINUE

        GOTO 1150

C       NAMELIST ENTRY SEGMENT

1000    CALL NULST(INLNM,1,8)
        DO 1020 I=1,6
1020    INLNM(I)=IDFILE(I+3,1)
1040    CALL INPTS(ITXT,INLNM,8,$8180,$9500)
        DO 1060 I=4,9
1060    INLDSC(I)=INLNM(I-3)
        INLDSC(1)="D<0>"
        INLDSC(2)=0
        INLDSC(3)=1
        INLDSC(10)="NL"
        INLDSC(11)=0
        CALL FSLKU(INLDSC,$1080,$1080,IER)
        GOTO 1100

1080    WRITE(TTO,8200) IER
        GOTO 1040

1100    IC=0
        IER=22K
        CALL FMOPS(INLDSC,ICH,$1080,IER)
        CALL FMRD(IS,ICH,11*MAXSM,$1120,IEE)
        IEE=0
1120    CALL FMCLS(ICH,$1140)
1140    IF(IEE.GT.0) GOTO 1080
        IC=(11*MAXSM+IEE)/11
        NX=NX+IC

C       KEYBOARD ENTRY SEGMENT

1150    CALL NULST(IFLNM,1,8)
        IF(NX.GE.MAXSM) GOTO 1200
        WRITE(TTO,8220)
        READ(TTO,8240,END=1200) IFLNM
        CALL GOBBL(IFLNM,8,ILEN)
        IF(ILEN.LE.0) GOTO 1200
        NX=NX+1
        DO 1180 I=4,9
1180    IS(I,NX)=IFLNM(I-3)

        GOTO 1150
```

```
C=========================================================
C              NAMELIST VARIFICATION
C=========================================================

1200   WRITE(TTO) " REVIEW OF FILENAMES"

       ITYP=0
       NG=NO+1
       DO 1400 NI=NG,NX
          CALL NULST(IFLDSC,1,11)
          CALL NULST(IFLFN,1,10)
          DO 1220 I=4,9
             I1=I-3
             IFLFN(I1)=IS(I,NI)
             IFLDSC(I)=IS(I,NI)
1220      CONTINUE
          IFLDSC(1)="D<0>"
          IFLDSC(2)=0
          IFLDSC(3)=1
          IFLDSC(10)="RD"
          LN=LENG(IFLFN,10)
          LNN=LN+1
          CALL MAPST(IEXTR,1,3,2,IFLFN,10,LNN,$1260)
          CALL OPENR(IFLFN,ICH,IER,$1260)
          GOTO 1280

1260      CALL CLOSE(ICH,IE)
          IE=1
          GOTO 1360

1280      CALL NULST(ITMP,1,20)
          READ(ICH,8260,END=1260,ERR=1260) NIND,IACCNT,IPMTH,IANLST,
    1        IEXDT,IACDT,IPRDT,ICODE
          READ(ICH,8280,END=1260,ERR=1260) INST,KUNITS,VOL,VEX,VINJ,
    1        FACTOR,NCOMP,(ITMP(I),I=1,6)
          READ(ICH,8300,END=1260,ERR=1260) IRGBK,ICKST,(ITXT(I),I=1,21)
          READ(ICH,8320,END=1260,ERR=1260) ISMPLE
          READ(ICH,8320,END=1260,ERR=1260) ICOND
          READ(ICH,8320,END=1260,ERR=1260) ICMNT

          CALL CLOSE(ICH,IE)

          IE=2
          CALL NULST(ITMP,8,20)
          DO 1300 I=4,9
               IF(ITMP(I-3).NE.IGCDSC(I)) GOTO 1360
1300      CONTINUE

          IE=3

          IF((ITYP.NE.0).AND.(ITYP.NE.ICODE)) GOTO 1360
          IF(MODE.EQ.0.AND.ICODE.GT.3) GOTO 1360
          IF(MODE.EQ.1.AND.ICODE.NE.6) GOTO 1360
          IF(MODE.EQ.2.AND.ICODE.NE.4) GOTO 1360
          IF(MODE.EQ.3.AND.ICODE.NE.2) GOTO 1360
          IF(MODE.EQ.4.AND.ICODE.NE.6) GOTO 1360
          IF(MODE.EQ.5.AND.ICODE.EQ.5) GOTO 1360
          IF(MODE.EQ.6.AND.ICODE.NE.7) GOTO 1360
          IF(MODE.EQ.7.AND.IOODE.NE.7) GOTO 1360

          IF(ITYP.EQ.0) ITYP=ICODE
          WRITE(TTO,8400)
          WRITE(TTO,8420) (IFLDSC(I),I=1,11)

          N=N+1
          GOTO 1400

1360      WRITE(TTO,8440)
          IS(11,NI)="NO"
          WRITE(TTO,8420) (IFLDSC(I),I=1,11)
          IF(IE.EQ.1) WRITE(TTO) " <7>DATA FILE ERROR"
          IF(IE.EQ.2) WRITE(TTO) " <7>QC<57>MT ERROR"
          IF(IE.EQ.3) WRITE(TTO) " <7>SAMPLE TYPE ERROR"
```

```
1400    CONTINUE

        NO=NX

        IF(N.GE.3) GOTO 1480
        WRITE(TTO) " <7>NOT ENOUGH ACCEPTABLE NAMES"
        GOTO 620

C1480   CALL NULST(INLNM,1,8)
C       CALL NULST(INLDSC,1,11)
C       INLNM(1)="DM"
C       INLNM(2)="P<0>"
C       INLDSC(1)="D<0>"
C       INLDSC(3)=1
C       INLDSC(4)="DM"
C       INLDSC(5)="P<0>"
C       INLDSC(10)="NL"
C       CALL FMOPN(INLDSC,ICH,$9000,IEE)
C       CALL FMWRT(IS,ICH,11*NX,$1500,IEE)
C1500   CALL FMCLT(ICH,$9000)

1480    CALL LOADIT("DMPR2")

8000    FORMAT("1",//," REPLICATE ANALYSIS MODULE (DMPR1)",/)
8020    FORMAT("  GC<57>METHOD TEMPLATE NAME [",4A2,"] > ",Z)
8040    FORMAT(" <7>",/,
     1      " ANALYSIS OPTIONS:",/,
     1      "  U -CONCENTRATION FOR UNKNOWNS",/,
     1      "  N -CONCENTRATION FOR NON-SFC SAMPLES",/,
     1      "  C -CONC.,%RECOVERY,& MDLR FOR CONTROL STANDARDS",/,
     1      "  F -RECOVERED CONC.& %RECOVERY FOR SPIKED SAMPLES",/,
     1      "  M -CONC.& MDLX FOR MODIFIED MATRIX NON-SFC SAMPLES",/,
     1      "  T -REL.RET.TIME FOR ANY NON-BLANK SAMPLE TYPE",/,
     1      "  R -RESPONSE FACTOR FOR CALIBRATION STANDARDS",/,
     1      "  A -RESPONSE RATIO (AR/CN) FOR CAL.STANDARDS")
8080    FORMAT("  OPTION (UNCFMTRAH) [",5A2,"] >"Z)
8100    FORMAT(" <7>",/,
     1      " LIST DEFINITION MODES:",/,
     1      "  K -KEYBOARD ENTRY OF EACH FILENAME",/,
     1      "  L -NAMELIST PLUS KEYBOARD ENTRIES",/,
     1      "  W -WILDCARD PLUS KEYBOARD ENTRIES")
8140    FORMAT("  OPTIONS (KLWH) [",5A2,"] > ",Z)
8160    FORMAT("  ROOT WORD AND IMBEDDED WILD CARD [",20A2,"] > ",Z)
8180    FORMAT("  NAME OF LIST FILE [",8A2,"] > ",Z)
8200    FORMAT(" <7>NAMELIST NOT ACCESSIBLE - ERROR ",OI4)
8220    FORMAT("  DATA FILE NAME [CR TO EXIT] > ",Z)
8240    FORMAT(8A2)
8260    FORMAT(I4,6(4A2),I4)
8280    FORMAT(2(3A2),4F8.3,2A2,6A2)
8300    FORMAT(5A2,5A2,21A2)
8320    FORMAT(32A2)
8420    FORMAT(" FILE: ",A2,":",OI3," ",OI3,"-",6A2,".",A2,"  ",A2)
8400    FORMAT(" ACCEPTED ",Z)
8440    FORMAT(" <7>REJECTED ",Z)

9000    WRITE(TTO,9020) IEE
9020    FORMAT(" <7>FATAL ERROR ",OI4)

9500    CALL RESET
        CALL LOADIT("EXEC")
        END
```

```
C       DMPR2.FR  REPLICATE ANALYSIS MODULE - STATISTICAL COMPUTATIONS
C       D.RYAN 2/82

        PARAMETER TTO=10,LPT=2,MAXN=120,MAXSM=25,DSK=1,ICH=1,LCH=4,MCH=5

        COMMON /CJNK1/ INST(3),RESPF(MAXN),
     1  ISMPLE(32),IANLST(4),ICOND(32),IACCNT(4),ITMP(20),ITXT(50),
     1  IACDT(4),IACDT2(4),NAME(35),IPMTH(4),
     1  IEXDT(4),IEXDT2(4),ICKST(5),IRGBK(5),ICMNT(32),ICAS(6),NCOMP(2),
     1  IANS(5),IYR,IDAY,IMO,IX,MODEP,LMD,LN,LNN,
     1  NM,NO,NN,IC,ICHR,IPOS,ILEN,NIND,ICODE,X,ITP,
     1  LOOP,MI,IP,CN,AR,IDT,IRRT,F,ICD,M1,M2,I1,I2,INDQ,ICHAR,
     1  IND,IUSE,RMDL,XMDL,A,B,C,D,KEYD,RMEAN,SIGR,KEYR,XLSA2,
     1  SMEAN,SIGS,KEYS,XLCT,CMEAN,SIGC,KEYC,A1,A2,A3,XQC,SQC,YMDL

        COMMON /CJNK2/ IEXTC(2),IEXTR(2),IQST1(5),IQST2(3),
     1  T(24),ITYPS(32),KH1(36),KH2(36),KH3(36),KH1A(11),KH2A(11),
     1  KH3A(11),KH1B1(6),KH1B2(6),KH2B(6),KH3B(6),KH1C(3),KH2C(3),
     1  KH3C(3),ITEMS(64)

        COMMON /CJNK3/
     1  IQCDSC(11),IQCFN(10),IFLDSC(11),IFLNM(8),IFLFN(10),INLDSC(11),
     1  INLNM(8),AMAT(MAXN,3),KUNITS(3),IS(11,MAXSM),ITYP,MODE,M,NI,NQ,
     1  NX,NO,IH1,IM1,IS1,N,FMDL

        COMMON /CJNK4/
     1  IQCNM(4),IPRDT(4)

        COMMON /CMAIN/ NPROG,IDFSW,IDFSW1,ITEMP(15),
     1  IDFILE(12,6),ISFILE(11,4),INSTD(3),IREFTABLE(11),
     1  IADD1,IADD2,ISUB1,ISUB2,ISUB3,ISUB4,JADD1,
     1  LIBNAME,LIBNUMB,IBUF,ISCN(45),IACQ(100),IDAT(45)

        DATA IEXTC/".QC<0>"/
        DATA IEXTR/".RD<0>"/
        DATA IQST1/"UNCFMTRA<0>H"/
        DATA IQST2/"KLW<0><0>H"/

        DATA T/63.657,9.925,5.841,4.604,4.032,3.707,3.499,3.355,
     1          3.250,3.169,3.106,3.005,3.012,2.977,2.947,2.921,
     1          2.898,2.878,2.861,2.845,2.831,2.819,2.807,2.797/

        DATA ITYPS/"UNKNOWN DUPED UNSPIKED USPK<53>DUP ",
     1             "CNTRL STRGNT BLKNON-SFC CAL STND"/

        DATA KH1/ 16*20040K," CAS              AVERAGE   STANDARD  % REL  "/
        DATA KH2/ " NO. COMPOUND NAME",7*20040K,"NUMBER           VALUE ",
     1          "  DEVIATION  STDEV  "/
        DATA KH3/ 36*36475K/
        DATA KH1A/ "  KNOWN   MEAN  LIMIT "/
        DATA KH2A/ "  VALUE   %REC  RANGE "/
        DATA KH3A/ 11*36475K/
        DATA KH1B1/ "    REAGENT "/
        DATA KH1B2/ "     MATRIX "/
        DATA KH2B/ "      MDL   "/
        DATA KH3B/ 6*36475K/
        DATA KH1C/ "NO.OF "/
        DATA KH2C/ "ITEMS "/
        DATA KH3C/ "======"/
```

```
      DATA ITEMS/ "CONCENTRATION   CONCENTRATION   CONC,%REC,& MDLR",
     1            "REC.CONC,%REC   MATRIX MDL      REL.RET.TIME    ",
     1            "RESPONSE FACTOR RESPONSE RATIO  "/

      WRITE(TTO,8000)
      IF(N.LE.2) GOTO 9000

C     IC=0
C     IER=22K
C     CALL FMOPS(INLDSC,ICH,$9000,IER)
C     CALL FMRD(IS,ICH,11*MAXSM,$1120,IEE)
C     IEE=0
C1120 CALL FMCLS(ICH,$1140)
C1140 IF(IEE.GT.0) GOTO 9000
C     IC=(11*MAXSM+IEE)/11
C     NX=IC

C=========================================================
C               LOOP 1: MEAN VALUE SUMMATION
C               LOOP 2: STANDARD DEVIATION SUMMATION
C=========================================================

C     WRITE(TTO,1211) M,MODE,NX
C1211 FORMAT(" CANDIDATES: ",I4," MODE: ",I3, " NUMBER OF FILES: ",I3)
      DO 1200 I=1,MAXN
         AMAT(I,1)=0.
         AMAT(I,2)=0.
         AMAT(I,3)=0.
         RESPF(I)=0.
1200  CONTINUE

      MODEP=MODE+1

      DO 2000 LOOP=1,2

C=========================================================
C               FOR EACH ACCEPTABLE REPLICATE
C=========================================================

         DO 1880 NI=1,NX

            IF(IS(11,NI).EQ."NO") GOTO 1880

            CALL NULST(IFLDSC,1,11)
            CALL NULST(IFLFN,1,10)
            DO 1400 I=4,9
               I1=I-3
               IFLFN(I1)=IS(I,NI)
               IFLDSC(I)=IS(I,NI)
1400        CONTINUE
            IFLDSC(1)="D<0>"
            IFLDSC(2)=0
            IFLDSC(3)=1
            IFLDSC(10)="RD"
            LN=LENG(IFLFN,10)
            LNN=LN+1
            CALL MAPST(IEXTR,1,3,2,IFLFN,10,LNN,$1420)
            IF(LOOP.EQ.1) WRITE(TTO,8020) (IFLDSC(I),I=1,11)
            CALL OPENR(IFLFN,ICH,IER,$1420)
            GOTO 1430

1420        WRITE(TTO) " <7>REDUCED DATA FILE I<57>O ERROR"
            GOTO 1860

1430        READ(ICH,8040,END=1420,ERR=1420) NIND,IACCNT,IPMTH,IANLST,
     1         IEXDT,IACDT,IPRDT,ICODE
            READ(ICH,8060,END=1420,ERR=1420) INST,KUNITS,VOL,VEX,VINJ,
     1         FACTOR,NCOMP,(ITMP(I),I=1,6)
            READ(ICH,8080,END=1420,ERR=1420) IRGBK,ICKST,(ITXT(I),I=1,13)
            READ(ICH,8100,END=1420,ERR=1420) ISMPLE
            READ(ICH,8100,END=1420,ERR=1420) ICOND
            READ(ICH,8100,END=1420,ERR=1420) ICMNT
            M=NIND
```

```
C=========================================================
C              GET INTERNAL STANDARD CN/AR'S
C=========================================================

               IF(MODE. NE. 6) GOTO 1500
               DO 1460 I=1, M
                   READ(ICH, 8120, END=1420, ERR=1420) IP, CN, AR, IDT, IPRT
                   IF(IP. NE. IDT) GOTO 1460
                   IF(AR. LT. 0. 5) GOTO 1460
                   RESPF(I)=CN/AR
1460           CONTINUE
               CALL POSITION(ICH, 0, 0, 371, IER, $1420)

C========================================================
C              FOR EACH ANALYTE IN FIXED LIST
C========================================================

1500           DO 1740 MI=1, M
                   READ(ICH, 8120, END=1420, ERR=1420) IP, CN, AR, IDT, IRRT
                   X=0. 0

C                         U     N     C     F     M     T     R     A
1580               GOTO (1600, 1600, 1600, 1660, 1600, 1640, 1630, 1620) MODEP

1600               X=CN
                   GOTO 1700

1620               IF(CN. LT. 0. 0005) GOTO 1680
                   X=AR/CN
                   GOTO 1700

1630               IF(CN. LT. 0. 0005) GOTO 1680
                   X=(RESPF(IDT)*AR)/CN
                   GOTO 1700

1640               X=FLOAT(IRRT)/1000.
                   GOTO 1700

1660               X=-CN

                   IF(LOOP. EQ. 1) GOTO 1670
                   RESPF(MI)=X
                   GOTO 1740

1670               AMAT(MI, 1)=AMAT(MI, 1)+X
                   AMAT(MI, 3)=AMAT(MI, 3)+1. 0
                   GOTO 1740

1680               WRITE(TTO, 8160) MI
                   GOTO 1740

1700               IF(LOOP. EQ. 1) GOTO 1670

1720               AMAT(MI, 2)=AMAT(MI, 2)+(X-AMAT(MI, 1))**2

1740           CONTINUE

               IF(MODE. NE. 3) GOTO 1860
```

```
C============================================================
C               FOR SPIKED SAMPLES (REMAINDER OF .RD)
C============================================================

1750            READ(ICH,8140,END=$1850,ERR=$1850) ICD,IEXDT2,
     1              IACDT2,IPRDT2,VOL,VEX,VINJ,FACTOR
                IF(ICD.EQ.2) GOTO 1800

C               BYPASS DUPLICATE INFO IF PRESENT

                IF(ICD.NE.1) GOTO 1850
                DO 1780 I=1,M
                    READ(ICH,8120,END=$1850,ERR=$1850) IP,CN,AR,IDT,IRRT
1780            CONTINUE
                GOTO 1750

1800            DC 1840 MI=1,M
                    READ(ICH,8120,END=$1850,ERR=$1850) IP,CN,AR,IDT,IRRT
                    IF(IP.EQ.IDT) GOTO 1840
                    IF(LOOP.EQ.1) GOTO 1820
                    AMAT(MI,2)=AMAT(MI,2)+(CN+RESPF(MI)-AMAT(MI,1))**2
                    GOTO 1840
1820                AMAT(MI,1)=AMAT(MI,1)+CN
1840            CONTINUE
                GOTO 1860

1850            WRITE(TTO) " <7>SPIKE PROCESSING ERROR"

1860            CALL RESET

1880        CONTINUE

C============================================================
C               MEAN OR STANDARD DEVIATION COMPUTATION LOOP
C============================================================

            DO 1980 MI=1,M
                IF(AMAT(MI,3).GT.2.0) GOTO 1940
                WRITE(TTO,8160) MI
                AMAT(MI,1)=0.0
                AMAT(MI,2)=0.0
                GOTO 1980
1940            IF(LOOP.EQ.1) GOTO 1960
                AMAT(MI,2)=SQRT(AMAT(MI,2)/(AMAT(MI,3)-1))
                GOTO 1980
1960            AMAT(MI,1)=AMAT(MI,1)/AMAT(MI,3)
1980        CONTINUE


2000    CONTINUE

        DO 2100 I=1,M
            WRITE(TTO,2020) AMAT(I,1),AMAT(I,2),AMAT(I,3)
2020        FORMAT("  MEAN: ",F12.3,"   STD.DEV: ",F12.3,"   REPS: ",F8.2)
2100    CONTINUE

        CALL RESET
        CALL LOADIT("DMPR3")
```

```
8000    FORMAT("1",//," STATISTICAL COMPUTATIONS (DMPR2)",/)
8020    FORMAT(" PROCESSING FILE: ",A2,":",OI3," ",OI3,"-",
     1  6A2,".",A2," ",A2)
8040    FORMAT(I4,6(4A2),I4)
8060    FORMAT(2(3A2),4F8.3,2A2,6A2)
8080    FORMAT(5A2,5A2,21A2)
8100    FORMAT(32A2)
8120    FORMAT(I4,F12.3,F12.0,I5,I6)
8140    FORMAT(I4,3(4A2),4F8.3)
8160    FORMAT(" COMPUTATION ERROR ON COMPOUND ",I3,". WILL CONTINUE.")

9000    WRITE(TTO,9020) IEE
9020    FORMAT(" <7>FATAL ERROR: ",OI4)
        CALL RESET
        WRITE(TTO) " REPLICATE ROUTINE TERMINATING"
        CALL LOADIT("EXEC")
        END
```

```
C       DMPR3.FR  REPLICATE ANALYSIS MODULE - STATISTICAL REPORT GENERATOR
C       D.RYAN 2/82     LAST REVISION: 09/23/82

        PARAMETER TTO=10,LPT=2,MAXN=120,MAXSM=25,DSK=1,ICH=1,LCH=4,MCH=5

        COMMON /CJNK1/ INST(3),RESPF(MAXN),
     1  ISMPLE(32),IANLST(4),ICOND(32),IACCNT(4),ITMP(20),ITXT(50),
     1  IACDT(4),IACDT2(4),NAME(35),IPMTH(4),
     1  IEXDT(4),IEXDT2(4),ICKST(5),IRGBK(5),ICMNT(32),ICAS(6),NCOMP(2),
     1  IANS(5),IYR,IDAY,IMO,IX,MODEP,LMD,LN,LNN,
     1  NM,NO,NN,IC,ICHR,IPOS,ILEN,NIND,ICODE,X,ITP,
     1  LOOP,MI,IP,CN,AR,IDT,IRRT,F,ICD,M1,M2,I1,I2,INDQ,ICHAR,
     1  IND,IUSE,RMDL,XMDL,A,B,C,D,KEYD,RMEAN,SIGR,KEYR,XLSA2,
     1  SMEAN,SIGS,KEYS,XLCT,CMEAN,SIGC,KEYC,A1,A2,A3,XGC,SGC,YMDL

        COMMON /CJNK2/ IEXTC(2),IEXTR(2),IGST1(5),IGST2(3),
     1  T(24),ITYPS(32),KH1(36),KH2(36),KH3(36),KH1A(11),KH2A(11),
     1  KH3A(11),KH1B1(6),KH1B2(6),KH2B(6),KH3B(6),KH1C(3),KH2C(3),
     1  KH3C(3),ITEMS(64)

        COMMON /CJNK3/
     1  IGCDSC(11),IGCFN(10),IFLDSC(11),IFLNM(8),IFLFN(10),INLDSC(11),
     1  INLNM(8),AMAT(MAXN,3),KUNITS(3),IS(11,MAXSM),ITYP,MODE,M,NI,NQ,
     1  NX,NO,IH1,IM1,IS1,N,FMDL

        COMMON /CJNK4/
     1  IGCNM(4),IPRDT(4)

        COMMON /CMAIN/ NPROG,IDFSW,IDFSW1,ITEMP(15),
     1  IDFILE(12,6),ISFILE(11,4),INSTD(3),IREFTABLE(11),
     1  IADD1,IADD2,ISUB1,ISUB2,ISUB3,ISUB4,JADD1,
     1  LIBNAME,LIBNUMB,IBUF,ISCN(45),IACG(100),IDAT(45)

        DATA IEXTC/".QC<0>"/
        DATA IEXTR/".RD<0>"/
        DATA IGST1/"UNCFMTRA<0>H"/
        DATA IGST2/"KLW<0><0>H"/

        DATA T/63.657,9.925,5.841,4.604,4.032,3.707,3.499,3.355,
     1          3.250,3.169,3.106,3.005,3.012,2.977,2.947,2.921,
     1          2.898,2.878,2.861,2.845,2.831,2.819,2.807,2.797/

        DATA ITYPS/"UNKNOWN DUPED UNSPIKED USPK<53>DUP ",
     1             "CNTRL STRGNT BLKNON-SFC CAL STND"/

        DATA KH1/ 16*20040K," CAS           AVERAGE   STANDARD  % REL  "/
        DATA KH2/ " NO.  COMPOUND NAME",7*20040K,"NUMBER          VALUE ",
     1         "  DEVIATION  STDEV  "/
        DATA KH3/ 36*36475K/
        DATA KH1A/ "  KNOWN   MEAN  LIMIT "/
        DATA KH2A/ "  VALUE   %REC  RANGE "/
        DATA KH3A/ 11*36475K/
        DATA KH1B1/ "    REAGENT "/
        DATA KH1B2/ "     MATRIX "/
        DATA KH2B/ "      MDL   "/
        DATA KH3B/ 6*36475K/
        DATA KH1C/ "NO.OF "/
        DATA KH2C/ "ITEMS "/
        DATA KH3C/ "======"/

        DATA ITEMS/ "CONCENTRATION   CONCENTRATION   CONC,%REC.& MDLR",
     1              "REC.CONC,%REC   MATRIX MDL      REL.RET.TIME    ",
     1              "RESPONSE FACTOR RESPONSE RATIO  "/

        MODEP=MODE+1

        WRITE(TTO) " REPORT NOW BEING GENERATED"
```

```
C       IC=0
C       IER=22K
C       CALL FMOPS(INLDSC, ICH, $9000, IER)
C       CALL FMRD(IS, ICH, 11*MAXSM, $200, IEE)
C       IEE=0
C200    CALL FMCLS(ICH, $300)
C300    IF(IEE.GT.0) GOTO 9000
C       IC=(11*MAXSM+IEE)/11
C       NX=IC

        CALL FSLKU(IGCDSC, $9000)
        CALL OPENR(IGCFN, LCH, IER, $500)
        GOTO 900

500     WRITE(TTO, 8040) IER
        CALL RESET
        GOTO 9500

900     READ(LCH, 8060, END=500, ERR=500) (ITMP(I), I=1, 7), NM, NO, NN,
     1      (ITXT(I), I=1, 16)

        CALL DELETE("1:DMPRPTR.99", IER, $1020)
1020    CALL OPENW("1:DMPRPTR.99", MCH, IER, $1970)

C=========================================================
C                WRITE REPORT HEADER
C=========================================================

        WRITE(MCH, 8080)
        M1=MODE*8+1
        M2=8*(MODE+1)
        WRITE(MCH, 8100) (ITEMS(I), I=M1, M2),
     1  IGCDSC(1), (IGCDSC(I), I=4, 10)
        I1=ITYP*4+1
        I2=4*(ITYP+1)
        WRITE(MCH, 8120) (ITYPS(I), I=I1, I2), (KUNITS(I), I=1, 3)
        WRITE(MCH, 8140) N, (IPRDT(I), I=1, 4)


C=========================================================
C                SHOW SAMPLE NAMES
C=========================================================

        WRITE(MCH, 8160)
        I1=0
        DO 1140 J=1, NX
            IF(IS(11, J).EQ."NO") GOTO 1140
            I1=I1+1
            WRITE(MCH, 8180) (IS(I, J), I=4, 9)
            IF(MOD(I1, 5).EQ.0) WRITE(MCH, 8200)
1140    CONTINUE

C=========================================================
C                WRITE SUBHEADER - LINE 1
C=========================================================

        IF(MOD(I1, 5).NE.0) WRITE(MCH, 8200)
        WRITE(MCH, 8200)
        WRITE(MCH, 8220) (KH1(I), I=1, 36)

C             U     N     C     F     M     T     R     A
        GOTO(1200, 1200, 1220, 1220, 1300, 1320, 1200, 1200) MODEP

1200    WRITE(MCH, 8200)
        GOTO 1360

1220    WRITE(MCH, 8240) (KH1A(I), I=1, 11)
        IF(MODE.NE.2) GOTO 1200
        WRITE(MCH, 8260) (KH1B1(I), I=1, 6)
        GOTO 1200

1300    WRITE(MCH, 8260) (KH1B2(I), I=1, 6)
        GOTO 1200

1320    WRITE(MCH, 8280) (KH1C(I), I=1, 3)
```

```
C========================================================
C                 WRITE SUBHEADER - LINE 2
C========================================================


1360    WRITE(MCH,8220) (KH2(I) I=1,36)

C                 U    N    C    F    M    T    R    A
        GOTO (1380,1380,1400,1400,1410,1420,1380,1380) MODEP

1380    WRITE(MCH,8200)
        GOTO 1440

1400    WRITE(MCH,8240) (KH2A(I),I=1,11)
        IF(MODE.NE.2) GOTO 1380
1410    WRITE(MCH,8260) (KH2B(I),I=1,6)
        GOTO 1380

1420    WRITE(MCH,8280) (KH2C(I),I=1,3)

C========================================================
C                 WRITE SUBHEADER - LINE 3
C========================================================

1440    WRITE(MCH,8220) (KH3(I),I=1,36)

C                 U    N    C    F    M    T    R    A
        GOTO (1460,1460,1480,1480,1500,1520,1460,1460) MODEP

1460    WRITE(MCH,8200)
        GOTO 1580

1480    WRITE(MCH,8240) (KH3A(I),I=1,11)
        IF(MODE.NE.2) GOTO 1460
1500    WRITE(MCH,8260) (KH3B(I),I=1,6)
        GOTO 1460
1520    WRITE(MCH,8280) (KH3C(I),I=1,3)

C========================================================
C                 WRITE MAIN BODY OF THE REPORT
C========================================================

1580    DO 1960 MI=1,M

            READ(LCH,8300,END=1870,ERR=1870) INDQ,(NAME(I),I=1,35)
            READ(LCH,8320,END=1870,ERR=1870) (ICAS(I),I=1,6),IUSE,
     1          RMDL,XMDL,A,B,C,D,KEYD
            READ(LCH,8340,END=1870,ERR=1870) RMEAN,SIGR,KEYR,XLSA2,SMEAN,
     1      SIGS,KEYS,XLCT,CMEAN,SIGC,KEYC

            IRP=INT(AMAT(MI,3)+0.5)
            IF(IRP.LE.2) GOTO 1960
            IF(IUSE.NE.3) GOTO 1660
            IF(MODE.LT.5) GOTO 1960
            ICHAR=44400K            ; "I<0>"
            CALL MOVBT(ICHAR,ICAS,6,1,$1700)
            GOTO 1700

1660        IF(IUSE.NE.2) GOTO 1680
            ICHAR=51400K            ; "S<0>"
            CALL MOVBT(ICHAR,ICAS,6,1,$1680)

1680        A3=0.0
            IF(MODE.NE.2.AND.MODE.NE.3) GOTO 1700
            IF(MODE.EQ.2) A3=XLCT
            IF(MODE.EQ.3) A3=XLSA2
            IF(A3.LT.0.0005) GOTO 1960

1700        RSD=0.0
            IF(AMAT(MI,1).LT.0.001) GOTO 1720
            RSD=AMAT(MI,2)*100.0/AMAT(MI,1)+0.05                  ;***
```

```
1720        A1=AMAT(MI, 1)+0. 0005
            A2=AMAT(MI, 2)+0. 0005
            WRITE(MCH, 8360) INDQ, (NAME(I), I=1, 12), (ICAS(I), I=1, 6),
     1          A1, A2, RSD

C                    U    N    C    F    M    T    R    A
            GOTO (1900, 1900, 1800, 1800, 1840, 1760, 1900, 1900) MODEP

1760        WRITE(MCH, 8380) IRP
            GOTO 1900

1800        XGC=AMAT(MI, 1)*100. 0/A3+0. 05                    ;***
            SGC=AMAT(MI, 2)*100. 0/A3+0. 05                    ;***
            A3=A3+0. 005
            WRITE(MCH, 8400) A3, XGC, SGC

            IF(MODE. NE. 2) GOTO 1900

1840        YMDL=T(IRP-1)*AMAT(MI, 2)+0. 0005
            MDLCH="<55><0>"
            IF(FMDL*YMDL. GE. AMAT(MI, 1)) MDLCH="<53><0>"
            WRITE(MCH, 8420) YMDL, MDLCH
            GOTO 1900

1870        WRITE(TTO, 8440)
            GOTO 1970

1900        DO 1910 I=13, 35
                IF(NAME(I). NE. "  ") GOTO 1930
1910        CONTINUE
            WRITE(MCH, 8200)
            GOTO 1960


1930        WRITE(MCH, 8460) (NAME(I), I=13, 35)

1960    CONTINUE

1970    WRITE(MCH, 8470)
        CALL RESET

        KCHAN=LPT
        CALL OPLPT(KCHAN, $2020)

        CALL OPENR("1:DMPRPTR. 99", DSK, IEE, $9000)
        CALL FCOPY(DSK, KCHAN, $2000, IE, IER)
2000    CALL CLOSE(DSK, IE)

        CALL CLOSE(KCHAN, IEE)
        GOTO 3000

2020    WRITE(TTO, 8480)

3000    CALL LOADIT("DMPR4")
```

```
8040    FORMAT(" <7>GC<57>MT ERROR: ",OI4)
8060    FORMAT(7A2,3I4,16A2,1X)
8080    FORMAT(" ",/,T20,"GC<57>MS REPLICATE STATISTICS REPORT",//)
8100    FORMAT("     ITEM OF INTEREST: ",8A2,T40,
     1      "GC METHOD TEMPLATE: ",A2,":",6A2,".",A2)
8120    FORMAT("     SAMPLE TYPE: ",4A2,T40,
     1       "UNITS OF CONC: ",3A2)
8140    FORMAT("     NUMBER OF REPLICATES: ",I4,T40,
     1      "DATE COMPUTED: ",4A2)
8160    FORMAT(" ",//,T25,"SAMPLE IDENTIFIERS USED",/)
8180    FORMAT(" ",6A2,T20,Z)
8200    FORMAT(" ")
8220    FORMAT(" ",36A2,Z)
8240    FORMAT(" ",11A2,Z)
8260    FORMAT(" ",6A2,Z)
8280    FORMAT(" ",3A2)
8300    FORMAT(I3,1X,35A2,2X)
8320    FORMAT(6A2,1X,I3,1X,2F7.2,1X,4F10.3,A1,1X)
8340    FORMAT(2F7.2,A1,2(F8.2,2F6.1,A1),1X)
8360    FORMAT(I4,1X,12A2,1X,6A2,F12.3,F10.3,1X,F6.1,1X,Z)
8380    FORMAT("  ",I4,Z)
8400    FORMAT(F9.2,1X,2(F6.1,1X),Z)
8420    FORMAT(F12.3,2X,A2,Z)
8440    FORMAT(" <7>GC<57>MT READ ERROR.  TERMINATING")
8460    FORMAT(" ",/,6X,23A2)
8470    FORMAT("1")
8480    FORMAT(" <7>LINE PRINTER NOT AVAILABLE.",/
     1          "<7>PLEASE GENERATE DMPRPTR.99 ASAP")

9000    WRITE(TTO,9020) IEE
9020    FORMAT(" <7>FATAL ERROR: ",OI4)

9500    CALL RESET
        WRITE(TTO) " REPLICATE ROUTINE TERMINATING"
        CALL LOADIT("EXEC")
        END
```

```
C        DMPP1.FR     DATA MANAGEMENT RUN RESULTS FILE PACKAGER
C        [illegible] RYAN 6/82        LAST REVISION: 8/23/83 FOR LCM/LRB FIX
C                                     11/21/83 FOR TYPE 7 RECORD CHANGE

         PARAMETER TTO=10, TTI=10, LPT=2, DSK=1
         PARAMETER MAXN=120, MAXSM=25

         COMMON /CUNK1/
     1   IRPDT(4), KVOL(2), IFORM(10), ILBDT(4), INST(3),
     1   ISMPLE(32), IANLST(4), ICOND(32), IACCNT(4), IPVTH(4),
     1   ITMP(20), ITXT(30), IACDT(4), IACDT2(4), KUNITS(3), NAME(35),
     1   METH(2), IPMTH(4), IEXDT(4), IEXDT2(4), ICKST(5),
     1   IRGDK(5), ICMNT(32), NCOMP(2), ICAS(6), ICOUNT(75,2), IQADT(4),
     1   NIND, ISAMP, IYR, IMO, IDAY, IHR, IMIN, ISEC, IX, ICP, ITYPE, ICHK, ICD,
     1   ICNTEE, ICR, NSAMP, ICODE, NPIND, IANS, I, I1, I2, NM, NO, NN, INDQ, IUSE,
     1   VOL, VEX, VINJ, FACTOR, CONC, D1, D2, D3, D4, X1, X2, X3, X4, X5, X7, IDT, IRRT,
     1   RMDL, AMDL, A, B, C, D, KEYD, RMEAN, SIGR, KEYR, XLSA2, SMEAN, SIGS, KEYS, AR,
     1   XLCT, CMEAN, SIGC, KEYC, IPMCT, ICH, LCH, MCH, ISPEC, MW, ID, LN, LNN,
     1   ITOTAL, I6CNT, I7CNT, I2CNT, ISCNT, ICCNT, IFCNT, IQBKS, ISMPLS

         COMMON /CUNK2/
     1   MCODE(4), IEXTC(2), IEXTX(2), IEXTR(2), IQCC(12), INIT(2), IRRCH(9),
     1   IRCAS(5), IQCT1(2), IQCT2(2), IQCT3(2), IQCT4(2), ISTRT(4), ISTOP(4)

         COMMON /CUNK3/
     1   IQCDSC(11), IQCFN(10), IFLDSC(11), IFLNM(8), IFLFN(10),
     1   INLDSC(11), INLNM(8), LSTMD, IRRNM(9), IRRDT(4), IQLNM, IS(11,MAXSM),
     1   IQCLS(10,20)

         COMMON /CUNK4/
     1   [illegible](4), IPRDT(4)

         COMMON /CMAIN/
     1   NPROG, IDFSW, IDFSW1, ITEMP(15),
     1   IDFILE(12,6), ISFILE(11,4),
     1   INSTD(3), IREFTABLE(11),
     1   IADD1, IADD2, ISUB1, ISUB2, ISUB3, ISUB4,
     1   JADD1, LIBRNAME, LIBRNUMB, IBUF,
     1   ISCN(45), IACQ(100), IDAT(45)

         DATA IQCC/"    LD1 LSO LDX LCM LRB "/
         DATA IRRCH/"RE","SU","LT","S ","FI","LE","NA","ME","  "/
         DATA MCODE/"UD","FX","CR","NS"/
         DATA IEXTC/".Q","C<O>"/
         DATA IEXTX/".R","R<O>"/
         DATA IEXTR/".R","D<O>"/
         DATA IQCT3/"LS1 "/
         DATA IQCT4/"LS2 "/

         D1=0.5
         D2=0.05
         D3=0.005
         D4=0.0005
         IQLNM=0
         ICHAN=1
         NIND=0
         ISCNT=0
         IQBKS=0
         ISAMP=0
         NSAMP=0
         ICH=3
         LCH=4
         MCH=5
         DO 10 I=1,10
             DO 10 J=1,20
                  IQCLS(I,J)=0
```

Reproduced from best available copy.

```
10      CONTINUE
        DO 20 I=1,75
            ICOUNT(I,1)=0
            ICOUNT(I,2)=0
20      CONTINUE
        CALL NULST(ITMP,1,20)
        CALL NULST(ITXT,1,50)
        CALL NULST(IRRNM,1,9)

        CALL GETTI(IYR,IMO,IDAY,IHR,IMIN,ISEC,$50,IX,IX,IX,IX,IX,IX)
        IRRDT(1)=(IMO/10+48)*256+MOD(IMO,10)+48
        IRRDT(2)=47*256+IDAY/10+48
        IRRDT(3)=(MOD(IDAY,10)+48)*256+47
        IRRDT(4)=(MOD(IYR/10,10)+48)*256+MOD(IYR,10)+48
        IRRNM(1)=(MOD(IYR/10,10)+48)*256+MOD(IYR,10)+48
        IRRNM(2)=(IMO/10+48)*256+MOD(IMO,10)+48
        IRRNM(3)=(IDAY/10+48)*256+MOD(IDAY,10)+48
        IRRNM(4)=(IHR/10+48)*256+MOD(IHR,10)+48
        IRRNM(5)=(IMIN/10+48)*256+MOD(IMIN,10)+48
        IRRNM(6)=".R"
        IRRNM(7)="R<0>"
        GOTO 140

50      WRITE(TTO) " <7>PLEASE SET TIME AT MSDS."
        CALL LOADIT("EXEC")

C       SALUTATION
C
140     WRITE(TTO,8000) IRRNM

        ITMP(1)="IN"
        ITMP(2)="TS"
        CALL INPTS(INIT,ITMP,2,$8010,$9020)
        CALL PADST(INIT,2)

300     ITMP(1)="XX"
        ITMP(2)="XY"
        ITMP(3)="ZM"
        ITMP(4)="N<0>"
        CALL INPTS(IPMTH,ITMP,4,$8020,$9020)
        ILEN=LENG(IPMTH,4)
        IF(ILEN.EQ.7) GOTO 400
        WRITE(TTO) " <7>USE SEVEN CHARACTERS PLEASE"
        GOTO 300

400     CALL PADST(IPMTH,4)
        CALL INPTS(INST,INSTD,3,$8030,$9020)
        CALL PADST(INST,3)

C       WRITE TYPE "1" RECORD
C
        CALL NULST(ITXT,1,32)
        CALL PADST(ITXT,32)
        CALL MAPST(INST,1,6,3,ITXT,32,2,$9000)
        CALL MAPST(IPMTH,1,8,4,ITXT,32,8,$9000)
        CALL MAPST(INIT,1,3,2,ITXT,32,16,$9000)
        CALL OPENW(IRRNM,MCH,IER,$9000)
        WRITE(MCH,8040) (IRRNM(I),I=1,5),(ITXT(I),I=1,32)
        ITOTAL=1

C       GET MODE: MANUAL OR NAMELIST
1150    WRITE(TTO,8050)
        IANS="M<0>"
        CALL INPTS(LSTMD,IANS,1,$8060,$7200)
        IF(IANS.EQ."X<0>") GOTO 7100
        IF(IANS.NE."L<0>".AND.IANS.NE."M<0>") GOTO 1150

        IF(LSTMD.NE."M<0>") GOTO 1200

C       GET NEXT MANUALLY ENTERED NAME
C
        DO 1180 I=1,6
            IFLNM(I)=ISFILE(I+3,1)
1180    CONTINUE
        CALL INPTS(ITXT,IFLNM,8,$8120,$1150)
        GOTO 1590
```

```
C       GET NAMELIST AND READ
C
1200    CALL NULST(INLNM, 1, 8)
        DO 1210 I=1, 6
             INLNM(I)=IDFILE(I+3, 1)
1210    CONTINUE
        CALL INPTS(ITMP, INLNM, 8, $8080, $1150)
        DO 1230 I=4, 9
             INLDSC(I)=INLNM(I-3)
1230    CONTINUE
        INLDSC(1)="D<0>"
        INLDSC(2)=0
        INLDSC(3)=1
        INLDSC(10)="NL"
        INLDSC(11)=0
        CALL FSLKU(INLDSC, $1240, $1240, IEE)
        GOTO 1260

C       NAMELIST DOES NOT COMPUTE
1240    WRITE(TTO, 8100) IEE
        GOTO 1200

1260    IC=0
        CALL FMOPS(INLDSC, KCHAN, $1240, IER)
        CALL FMRD(IS, KCHAN, 11*MAXSM, $1270, IEE)
        IEE=0
1270    CALL FMCLS(KCHAN, $1280)
1280    IF(IEE. GT. 0) GOTO 1240
        IC=(11*MAXSM+IEE)/11
        NSAMP=IC

C       GET NEXT NAME FROM LIST
C
1500    CALL NULST(IFLNM, 1, 8)
        ISAMP=ISAMP+1
        IF(ISAMP. GT. NSAMP) GOTO 1150
        DO 1520 I=4, 9
             I1=I-3
             IFLNM(I1)=IS(I, ISAMP)
1520    CONTINUE

C       PREPARE FILENAME FOR PROCESSING
C
1590    CALL NULST(IFLFN, 1, 10)
        DO 1600 I=1, 6
             I1=I+3
             IFLDSC(I1)=IFLNM(I)
             IFLFN(I)=IFLNM(I)
1600    CONTINUE

        LN=LENG(IFLNM, 8)
        LNN=LN+1
        IFLDSC(10)="RD"
        CALL MAPST(IEXTR, 1, 3, 2, IFLFN, 10, LNN, $4210)

        CALL OPENR(IFLFN, ICH, IER, $4210)
        GOTO 4220
```

```
C       HANDLE REDUCED DATA FILE I/O ERROR
C
4210    WRITE(TTO,8200) IFLFN
4215    CALL CLOSE(ICH,IER,$4218)
4218    CALL CLOSE(LCH,IER,$7090)
        GOTO 7090

C       READ HEADER PARAMETERS
C
4220    CALL  NULST(ITMP,1,20)
        READ(ICH,8230,END=4210,ERR=4210) NIND,IACCNT,IPMTH,IANLST,
     1      IEXDT,IACDT,IPRDT,ICODE
        READ(ICH,8240,END=4210,ERR=4210) INST,KUNITS,VOL,VEX,VINJ,FACTOR,
     1      NCOMP,(ITMP(I),I=1,6)
        READ(ICH,8250,END=4210,ERR=4210) IRGBK,ICKST,IPVTH,
     1      (ITXT(I),I=1,13)
        READ(ICH,8260,END=4210,ERR=4210) ISMPLE
        READ(ICH,8260,END=4210,ERR=4210) ICOND
        READ(ICH,8260,END=4210,ERR=4210) ICMNT
        IF(ICODE.GE.6) GOTO 4215
C
C       MIGHT BE USED TO CHECK HEADER TO SEE IF RD IS IN ANOTHER RR
C
C       DO  4230 I=1,9
C           IF(ITXT(I).NE.IRRCH(I)) GOTO 4210
C4230   CONTINUE
        I1=ICODE*2+1
        I2=I1+1
        IGCT1(1)=IGCC(I1)
        IGCT1(2)=IGCC(I2)

C       PREPARE HEADER VARIABLES AND GET GC/MT FILENAME
C
        CALL GOBBL(ITMP,20,LTX)
        DO 4240 I=1,8
            IGCFN(I)=ITMP(I)
            IGCDSC(I+3)=ITMP(I)
 4240    CONTINUE
         IGCFN(9)=0
         IGCFN(10)=0
         CAL'. MAPST(IEXTC,1,3,2,IGCFN,10,8,$4260)
         IGCDSC(1)="D<0>"
         IGCDSC(2)=0
         IGCDSC(3)=1
         IGCDSC(10)="GC"
         IGCDSC(11)=0
         CALL FSLKU(IGCDSC,$4260,$4260,IER)
         CALL MAPST(KUNITS,4,6,3,KVOL,2,1,$4260)
         ICP=ICODE+1
         ITYPE=NCHAR(MCODE,4,ICP)
C        ICP FORCED TO 1 FOR PRIMARY UNKNOWN
         IF(ICP.LE.4) ICP=1
         ICHK=0
         X1=VOL+D4
         X2=VEX+D4
         X3=VINJ+D4
        X4=FACTOR+D4

        CALL OPENR(IGCFN,LCH,IER,$4260)
        READ(LCH,8270,END=4260,ERR=4260) (ITMP(I),I=1,7),NM,NO,NN,
     1      (ITXT(I),I=1,16)
        IF(NM.NE.NIND) GOTO 4260
        GOTO 4280

C       GC/MT ACCESS ERROR
C
4260    WRITE(TTO,8280) IGCFN
        GOTO 4210
```

```
C       LOOK THROUGH LIST TO SEE IF TEMPLATE HAS BEEN USED
C
4280    DO 4290 I=1, 10
            ITXT(I)=IGCFN(I)
4290    CONTINUE
        ITXT(10)=ITYPE
        IF(ITYPE.EQ."X<0>") ITXT(10)="U<0>"
        IF(IGLNM.EQ.0) GOTO 4330
        DO 4320 I=1, IGLNM
            DO 4300 J=1, 10
                IF(ITXT(J).NE.IGCLS(J, I)) GOTO 4320
4300        CONTINUE
            GOTO 4360
4320    CONTINUE
4330    IF(IGLNM.EQ.20) GOTO 4260
        IGLNM=IGLNM+1
        DO 4340 I=1, 10
            IGCLS(I, IGLNM)=ITXT(I)
4340    CONTINUE
C       WRITE TYPE "6" RECORD FOR PRIMARY
C
4360    WRITE(TTO, 8300) (IFLDSC(I), I=4, 9)
        WRITE(TTO, 8310) IACCNT, ITYPE
        WRITE(TTO, 8320) KUNITS, (IGCDSC(I), I=4, 7)
        WRITE(TTO, 8330) X1, KVOL, X4
        WRITE(TTO, 8340) IEXDT, IACDT
        WRITE(TTO, 8350) X2, X3
        WRITE(TTO, 8360) IANLST, INST
        WRITE(TTO, 8370) IRGBK, ICKST
        ICCNT=0
        IFCNT=0
        I2CNT=0
        ISCNT=0
        CALL PADST(IFLNM, 8)
        WRITE(MCH, 8290) IACCNT, IPMTH, IFLNM
        WRITE(TTO) " NOW GENERATING RUN RESULTS FOR PRIMARY SAMPLE"
        I6CNT=I6CNT+1
        ITOTAL=ITOTAL+1
        ICOUNT(I6CNT, 2)=ITOTAL

C       READ QC/MT AND RD LINE ITEMS
C
4540    READ(LCH, 8540, END=4260, ERR=4260) INDG, (NAME(I), I=1, 35)
        READ(LCH, 8550, END=4260, ERR=4260) (ICAS(I), I=1, 6), IUSE, RMDL,
     1      XMDL, A, B, C, D, KEYD
        READ(LCH, 8560, END=4260, ERR=4260) RMEAN, SIGR, KEYR, XLSA2, SMEAN, SIGS,
     1      KEYS, XLCT, CMEAN, SIGC, KEYC

C       COMPRESS CAS NUMBER
        IRCAS(5)=ICAS(6)
        ICHAR=NCHAR(ICAS, 6, 10)
        CALL MOVBT(ICHAR, IRCAS, 6, 9, $4260)
        ICHAR=NCHAR(ICAS, 6, 9)
        CALL MOVBT(ICHAR, IRCAS, 6, 8, $4260)
        ICHAR=NCHAR(ICAS, 6, 7)
        CALL MOVBT(ICHAR, IRCAS, 6, 7, $4260)
        IRCAS(3)=ICAS(3)
        IRCAS(2)=ICAS(2)
        IRCAS(1)=ICAS(1)

        READ(ICH, 8570, END=4210, ERR=4210) IPMCT, CONC, AR, IDT, IRRT
        X5=CONC+D4
        IF(INDG.NE.IPMCT) GOTO 4260

C       FIRST TIME THROUGH D, F, &X NOT CONSIDERED.
C       SECOND TIME THROUGH ICP CAN ONLY BE 1 (D) OR 2 (F).
C
C ITYPE=>         U     D     F     X     C     R
        GOTO (4560, 4910, 4740, 9000, 5230, 5530) ICP
C ICODE=          0     1     2     3     4     5
```

```
C       OUTPUT UNKNOWN SAMPLE INFO
C
4560    IF(IUSE. EQ. 3) GOTO 4590
        IF(IUSE. EQ. 2) GOTO 4600
        WRITE(MCH, 8600) IACCNT, IGCT1, IRCAS, X5, KUNITS
        I2CNT=I2CNT+1

C       UNKNOWN AND DUPLICATE LINE TERMINATOR
C
4590    IF(INDG. NE. NIND) GOTO 4540
        ICOUNT(I6CNT, 1)=I2CNT-2*ISCNT
C       WRITE(TTO) I6CNT, ICOUNT(I6CNT, 1), ICOUNT(I6CNT, 2)
        ITOTAL=ITOTAL+I2CNT
        GOTO 4610

C       OUTPUT SURROGATE INFO FOR U, D, F, OR X
C
4600    ISCNT=ISCNT+1
        X7=XLSA2+D3
        WRITE(MCH, 8600) IACCNT, IGCT3, IRCAS, X5, KUNITS
        WRITE(MCH, 8600) IACCNT, IGCT4, IRCAS, X7, KUNITS
        I2CNT=I2CNT+2
        GOTO 4590

C       OUTPUT D OR F HEADER
C
4610    IF(ICODE. EQ. 0) GOTO 6000
        IF(ICHK. EQ. ICODE) GOTO 6000             ;ALL DONE WITH U, D, F, X
        CALL POSITION(LCH, 0, 0, 0, IER, $4260)
        READ(ICH, 8640, END=4210, ERR=4210) ICD, IEXDT2, IACDT2, IPRDT, VOL,
     1  VEX, VINJ, FACTOR
        ICHK=ICHK+ICD
        ICP=ICD+1

C       LOOK THROUGH LIST TO SEE IF TEMPLATE HAS BEEN USED
C
        DO 4620 I=1, 10
            ITXT(I)=IGCFN(I)
4620    CONTINUE
        ITXT(10)=NCHAR(MCODE, 4, ICP)
        IF(IGLNM. EQ. 0) GOTO 4650
        DO 4640 I=1, IGLNM
            DO 4630 J=1, 10
                IF(ITXT(J). NE. IGCLS(J, I)) GOTO 4640
4630        CONTINUE
            GOTO 4670
4640    CONTINUE
4650    IF(IGLNM. EQ. 20) GOTO 4260
        IGLNM=IGLNM+1
        DO 4660 I=1, 10
            IGCLS(I, IGLNM)=ITXT(I)
4660    CONTINUE

4670    IF(ICD. NE. 1) GOTO 4710
        IGCT1(1)="LD"
        IGCT1(2)="2 "
        IGCT2(1)="  "
        IGCT2(2)="  "
        GOTO 4720
4710    IF(ICD. NE. 2) GOTO 4210
        IGCT1(1)="LS"
        IGCT1(2)="F "
        IGCT2(1)="LS"
        IGCT2(2)="A "

C       WRITE TYPE "6" FOR DUPE OR SPIKE
C
4720    WRITE(MCH, 8290) IACCNT, IPMTH, IFLNM
        IF(ICD. EQ. 1) WRITE(TTO) " NOW GENERATING RUN RESULTS FOR DUPLICATE"
        IF(ICD. EQ. 2) WRITE(TTO) " NOW GENERATING RUN RESULTS FOR SPIKE"
        ITOTAL=ITOTAL+1
        I6CNT=I6CNT+1
        ICOUNT(I6CNT, 2)=ITOTAL
        ISCNT=0
```

```
      READ(LCH,8270,END=4260,ERR=4260) (ITMP(I),I=1,7),NM,NO,NN,
     1    (ITXT(I),I=1,16)
      GOTO 4540

C     OUTPUT SPIKE INFO
C
4740  IF(IUSE.EQ.3) GOTO 4760
      IF(IUSE.EQ.2) GOTO 4600
      IF(XLSA2.LE.D2) GOTO 4760
      IFCNT=IFCNT+1
      WRITE(MCH,8600) IACCNT,IGCT1,IRCAS,X5,KUNITS
      X7=XLSA2+D3
      WRITE(MCH,8600) IACCNT,IGCT2,IRCAS,X7,KUNITS
      I2CNT=I2CNT+2
4760  IF(INDG.NE.NIND) GOTO 4540
      ICOUNT(I6CNT,1)=IFCNT
C     WRITE(TTO) I6CNT,ICOUNT(I6CNT,1),ICOUNT(I6CNT,2)
      ITOTAL=ITOTAL+I2CNT
      GOTO 4610

C     OUTPUT DUPLICATE INFO
C
4910  IF(IUSE.EQ.3) GOTO 4590
      IF(IUSE.EQ.2) GOTO 4600
      WRITE(MCH,8600) IACCNT,IGCT1,IRCAS,X5,KUNITS
      I2CNT=I2CNT+1
      GOTO 4590

C     OUTPUT CONTROL STANDARD INFO
C
5230  IF(IUSE.EQ.3) GOTO 5240
      IF(XLCT.LE.D3) GOTO 5240
      ICCNT=ICCNT+1
      IGCT2(1)="LC"
      IGCT2(2)="T "
      WRITE(MCH,8600) IACCNT,IGCT1,IRCAS,X5,KUNITS
      X7=XLCT+D3
      WRITE(MCH,8600) IACCNT,IGCT2,IRCAS,X7,KUNITS
      I2CNT=I2CNT+2
5240  IF(INDG.NE.NIND) GOTO 4540
      ICOUNT(I6CNT,1)=ICCNT
C     WRITE(TTO) I6CNT,ICOUNT(I6CNT,1),ICOUNT(I6CNT,2)
      ITOTAL=ITOTAL+I2CNT
      GOTO 7000

C     OUTPUT REAGENT BLANK INFO
C
5530  IF(IUSE.EQ.3) GOTO 5540
      WRITE(MCH,8600) IACCNT,IGCT1,IRCAS,X5,KUNITS
      I2CNT=I2CNT+1
5540  IF(INDG.NE.NIND) GOTO 4540
      ICOUNT(I6CNT,1)=I2CNT
C     WRITE(TTO) I6CNT,ICOUNT(I6CNT,1),ICOUNT(I6CNT,2)
      ITOTAL=ITOTAL+I2CNT
      GOTO 7000

C     MANAGE QUALITATIVE OUTPUT HERE
C
6000  READ(ICH,8680,END=7000,ERR=7000) ICD,(ITXT(I),I=1,32)
      IF(ICD.NE.9) GOTO 7000
      I7CNT=0
      I2CNT=0
      IGCT1(1)="  "
      IGCT1(2)="  "
      WRITE(MCH,8290) IACCNT,IPVTH,IFLNM
      WRITE(TTO) " NOW GENERATING RUN RESULTS FOR QUALITATIVE DATA"
      I6CNT=I6CNT+1
      ITOTAL=ITOTAL+1
      IGBKS=IGBKS+1
      ICOUNT(I6CNT,2)=ITOTAL
```

```
6100    READ(ICH,8700,END=6800,ERR=6800) ISPEC,(NAME(I),I=1,34)
        READ(ICH,8710,END=6800,ERR=6800) CONC,ICAS,IFORM,MW,ID
        IF(ID.EQ."U ") GOTO 6100                  ;SKIP UNKNOWN PEAKS
        X5=CONC+D4

        IRCAS(5)=ICAS(6)
        ICHAR=NCHAR(ICAS,6,10)
        CALL MOVBT(ICHAR,IRCAS,6,9,$6800)
        ICHAR=NCHAR(ICAS,6,9)
        CALL MOVBT(ICHAR,IRCAS,6,8,$6800)
        ICHAR=NCHAR(ICAS,6,7)
        CALL MOVBT(ICHAR,IRCAS,6,7,$6800)
        IRCAS(3)=ICAS(3)
        IRCAS(2)=ICAS(2)
        IRCAS(1)=ICAS(1)
C       WRITE "7" TYPE RECORD
        I7CNT=I7CNT+1
        WRITE(MCH,8720) IACCNT,I7CNT,IRCAS,(NAME(I),I=1,26)
        IF(CONC.GT.D4) GOTO 6200
        WRITE(MCH,8740) IACCNT,IRCAS,KUNITS
        GOTO 6300
6200    WRITE(MCH,8600) IACCNT,IGCT1,IRCAS,X5,KUNITS
6300    I2CNT=I2CNT+1
        GOTO 6100

6800    ICOUNT(I6CNT,1)=I2CNT
C       WRITE(TTO) I6CNT,ICOUNT(I6CNT,1),ICOUNT(I6CNT,2)
        ITOTAL=ITOTAL+I7CNT+I2CNT

C       TAKE CARE OF REDUCED DATA FILE HEADER
C
7000    CALL CLOSE(ICH,IER,$9000)
        CALL CLOSE(LCH,IER,$9000)
        CALL OPEN(IFLFN,ICH,IER,$9000)
        CALL POSITION(ICH,0,0,149,IER,$9000)
        DO 7020 I=1,5
7020    ITXT(I)=IRRNM(I)
        DO 7040 I=6,9
7040    ITXT(I)="  "
        DO 7060 I=10,13
7060    ITXT(I)=IRRDT(I-9)
        WRITE(ICH,8760) (ITXT(I),I=1,13)
        CALL POSITION(ICH,1,-1,-1,IER,$7080)
7080    CALL CLOSE(ICH,IER,$9000)
7090    IF(LSTMD.EQ."L<0>") GOTO 1500
        GOTO 1150

C       PUT NUMBERS IN PLACE OF ###'S I "6" RECORDS
C
7100    CALL RESET
        ISMPLS=I6CNT-IGBKS
C       WRITE(TTO,8800) ITOTAL,ISMPLS,IGBKS,IGLNM
        IF(IGLNM.LE.0) GOTO 7190
        CALL OPEN(IRRNM,MCH,IER,$9000)
        DO 7120 I=1,I6CNT
            IREC=ICOUNT(I,2)-1
            INUM=ICOUNT(I,1)
C           WRITE(TTO) IREC,INUM
            CALL POSITION(MCH,IREC,81,11,IER,$7120)
            WRITE(MCH,8840) INUM
7120    CONTINUE
        CALL POSITION(MCH,1,-1,-1,IER,$7500)
        GOTO 9000

7190    WRITE(TTO) " <7>NO DATA AVAILABLE"

7200    CALL RESET
        CALL DELETE(IRRNM,IER,$9000)
        WRITE(TTO) " <7>RUN RESULTS FILE BEING PURGED."
        CALL LOADIT("DMPC")
        GOTO 9020
```

```
C7500   CALL NULST(ITMP,20)
C       ITMP(1)="N<0>"
C       CALL INPTS(IANS,ITMP,1,$8820,$9020)
C       IF(IANS.EQ."Y<0>") GOTO 9020
7500    CALL RESET
        CALL LOADIT("DMPP2")

8000    FORMAT(" ",/," RUN RESULTS FILE GENERATION MODULE (DMPP1)",/,
     1      " RUN RESULTS FILE NAME IS: ",9A2)
8010    FORMAT(" ENTER YOUR INITIALS [",2A2,"] > ",Z)
8020    FORMAT(" ENTER GENERIC PARMETH CODE [",4A2,"] > ",Z)
8030    FORMAT(" ENTER INSTRUMENT IDENTIFIER [",3A2,"] > ",Z)
8040    FORMAT(" 1",5(1X,A2),32A2)
8050    FORMAT("1 PROCESS MODES:",/,
     1          "    L - NAMELIST SAMPLE NAME ENTRY",/,
     1          "    M - MANUAL SAMPLE NAME ENTRY",/,
     1          "    X - EXIT TO SEND MODULE",/,
     1          "   ^D - EXIT WITHOUT ACTION")
8060    FORMAT("  L, M, OR X  [",A2,"] > ",Z)
8080    FORMAT(" NAME OF LIST FILE [",8A2,"] > ",Z)
8100    FORMAT(" <7>NAMELIST ACCESS ERROR: ",OI6)
8120    FORMAT(" SAMPLE NAME ($)  [",8A2,"] > ",Z)
C8140   FORMAT(" NOW REPORTING ON PRIMARY SAMPLE: ",6A2)
C8160   FORMAT(" NOW REPORTING ON DUPLICATE UNKNOWN")
C8180   FORMAT(" NOW REPORTING ON SPIKED UNKNOWN")
8200    FORMAT(" <7>UNABLE TO PROCESS REDUCED DATA FILE ",10A2)
8230    FORMAT(I4,6(4A2),I4)
8240    FORMAT(2(3A2),4F8.3,2A2,6A2)
8250    FORMAT(5A2,5A2,4A2,13A2)
8260    FORMAT(32A2)
8270    FORMAT(7A2,3I4,16A2,1X)
8280    FORMAT(" <7>UNABLE TO PROCESS GC<57>METHOD TEMPLATE FILE ",10A2)
8290    FORMAT(" 6 ",4A2," ### P    ",4A2,37X,8A2)
8300    FORMAT(1H1," RUN RESULTS NOW BEING GENERATED FOR SAMPLE: ",6A2)
8310    FORMAT("    SFC SAMPLE ID: "4A2,T38"SAMPLE TYPE: "A2)
8320    FORMAT("    UNITS: "3A2,T38"GC<57>MT NAME: "4A2)
8330    FORMAT("    SAMPLE VOLUME:      "F8.3" ",2A2,T38,
     1  "FACTOR:             "F8.3)
8340    FORMAT("    EXTRACTED: "4A2,T38"INJECTED: "4A2)
8350    FORMAT("    EXTRACTION VOLUME: "F8.3" ML"T38
     1  "INJECTION VOLUME: "F8.3" UL")
8360    FORMAT("    ANALYST: "4A2,T38"INSTRUMENT: "3A2)
8370    FORMAT("    REAGENT BLANK: ",5A2,T38,"CONTROL STANDARD: "5A2/)
8540    FORMAT(I3,1X,35A2,2X)
8550    FORMAT(6A2,1X,I3,1X,2F7.2,1X,4F10.3,A1,1X)
8560    FORMAT(2F7.2,A1,2(F8.2,2F6.1,A1),1X)
8570    FORMAT(I4,F12.3,F12.0,I5,I6)
8600    FORMAT(" 2 ",4A2,1X,2A2,"C",5A2,1X,E10.3,1X,3A2,36X)
8640    FORMAT(I4,3(4A2),4F8.3)
8680    FORMAT(I3,1X,32A2)
8700    FORMAT(I5,1X,34A2)
8710    FORMAT(18X,F12.3,1X,6A2,2X,10A2,I5,1X,A2)
8720    FORMAT(" 7 002 ",4A2,I4," C",5A2,1X,24A2,1X)
8740    FORMAT(" 2 ",4A2,"      C",5A2,12X,3A2,"   PNG",30X)
8760    FORMAT(1X,13A2,Z)
C8800   FORMAT(" ",/" RUN RESULTS FILE HAS",I6," RECORDS",/,
C    1      " WITH DATA FOR",I4," SAMPLES.",/,
C    1      " THERE ARE",I4," QUAL BLOCKS",/,
C    1      " AND",I4," DISTINCT PARTYPES USED.")
C8820   FORMAT(" ",/," EXIT TO MSDS [",A2,"] > ",Z)
8840    FORMAT(I4,Z)

9000    WRITE(TTO,9010) IER
9010    FORMAT(" <7>ERROR FATAL: ",OI4)
9020    CALL RESET
        CALL LOADIT("EXEC")

        END
```

```
C       DMPP2. FR     DATA MANAGEMENT RUN RESULTS FILE WRAPUP AND SEND
C       D. RYAN 7/82        LAST REVISION:  12/08/82

        PARAMETER TTO=10, TTI=10, LPT=2, DSK=1
        PARAMETER MAXN=120, MAXSM=25

        COMMON /CJNK1/
     1  IRPDT(4), KVOL(2), IFORM(10), ILBDT(4), INST(3),
     1  ISMPLE(32), IANLST(4), ICOND(32), IACCNT(4), IPVTH(4),
     1  ITMP(20), ITXT(50), IACDT(4), IACDT2(4), KUNITS(3), NAME(35),
     1  METH(2), IPMTH(4), IEXDT(4), IEXDT2(4), ICKST(5),
     1  IRGBK(5), ICMNT(32), NCOMP(2), ICAS(6), ICOUNT(75, 2), IGADT(4),
     1  NIND, ISAMP, IYR, IMO, IDAY, IHR, IMIN, ISEC, IX, ICP, ITYPE, ICHK, ICD,
     1  IC, IEE, IER, NSAMP, ICODE, NPIND, IANS, I, I1, I2, NM, NO, NN, INDG, IUSE,
     1  VOL, VEX, VINJ, FACTOR, CONC, D1, D2, D3, D4, X1, X2, X3, X4, X5, X7, IDT, IRRT,
     1  RMDL, XMDL, A, B, C, D, KEYD, RMEAN, SIGR, KEYR, XLSA2, SMEAN, SIGS, KEYS, AR,
     1  XLCT, CMEAN, SIGC, KEYC, IPMCT, ICH, LCH, MCH, ISPEC, MW, ID, LN, LNN,
     1  ITOTAL, I6CNT, I7CNT, I2CNT, ISCNT, ICCNT, IFCNT, IGBKS, ISMPLS

        COMMON /CJNK2/
     1  MCODE(4), IEXTC(2), IEXTX(2), IEXTR(2), IGCC(12), INIT(2), IRRCH(9),
     1  IRCAS(5), IGCT1(2), IGCT2(2), IGCT3(2), IGCT4(2), ISTRT(4), ISTOP(4)

        COMMON /CJNK3/
     1  IGCDSC(11), IGCFN(10), IFLDSC(11), IFLNM(8), IFLFN(10),
     1  INLDSC(11), INLNM(8), LSTMD, IRRNM(9), IRRDT(4), IGLNM, IS(11, MAXSM),
     1  IGCLS(10, 20)

        COMMON /CJNK4/
     1  IGCNM(4), IPRDT(4)

        COMMON /CMAIN/
     1  NPROG, IDFSW, IDFSW1, ITEMP(15),
     1  IDFILE(12, 6), ISFILE(11, 4),
     1  INSTD(3), IREFTABLE(11),
     1  IADD1, IADD2, ISUB1, ISUB2, ISUB3, ISUB4,
     1  JADD1, LIBRNAME, LIBRNUMB, IBUF,
     1  ISCN(45), IACQ(100), IDAT(45)

        DATA IGCC/" LCL LWL AVE UWL UCL"/
        DATA IRRCH/"RE", "SU", "LT", "S ", "FI", "LE", "NA", "ME', "   "/
        DATA MCODE/"UD", "FX", "CR", "NS"/
        DATA IEXTC/". G", "C<0>"/
        DATA IEXTX/". R", "R<0>"/
        DATA IEXTR/". R", "D<0>"/
        DATA IGCT3/"LS1 "/
        DATA IGCT4/"LS2 "/

        CALL NULST(ITMP, 1, 20)
        CALL NULST(ITXT, 1, 50)

C       WRITE(TTO, 2) IRRNM, IRRDT
C2      FORMAT(" RRNAME: ", 9A2, "    DATE: ", 4A2)
C       WRITE(TTO, 6) IGLNM
C6      FORMAT(" IGLNM: ", I6)
        DO 20 I=1, IGLNM
            WRITE(TTO, 10) (IGCLS(J, I), J=1, 10)
10          FORMAT(" PARMETH: ", 9A2, 5X, A2)
20      CONTINUE

        D1=0. 5
        D2=0. 05
        D3=0. 005
        D4=0. 0005
        KCHAN=1
        NIND=0
        ICH=3
        LCH=4
        MCH=5
        CALL GETTI(IYR, IMO, IDAY, IHR, IMIN, ISEC, $50, IX, IX, IX, IX, IX, IX)
        IPRDT(1)=(IMO/10+48)*256+MOD(IMO, 10)+48
        IPRDT(2)=47*256+IDAY/10+48
        IPRDT(3)=(MOD(IDAY, 10)+48)*256+47
        IPRDT(4)=(MOD(IYR/10, 10)+48)*256+MOD(IYR, 10)+48
        ITMP(1)=(MOD(IYR/10, 10)+48)*256+MOD(IYR, 10)+48
        ITMP(2)=(IMO/10+48)*256+MOD(IMO, 10)+48
        ITMP(3)=(IDAY/10+48)*256+MOD(IDAY, 10)+48
        ITMP(4)=(IHR/10+48)*256+MOD(IHR, 10)+48
        ITMP(5)=(IMIN/10+48)*256+MOD(IMIN, 10)+48
C       WRITE(TTO, 42) IPRDT, (ITMP(I), I=1, 5)
C42     FORMAT(" PRDATE:  ", 4A2, "     MATCHDT:  ", 5A2)
        GOTO 140

50      WRITE(TTO) " <7>PLEASE SET TIME AT MSDS"
        CALL LOADIT("EXEC")

C       FIND OUT IF FROM DMPP1 OR ELSEWHERE
C
140     IF(IRRNM(1). NE. ITMP(1)) GOTO 200                     ;YEAR
        IF(IRRNM(2). NE. ITMP(2)) GOTO 200                     ;MONTH
        IF(IRRNM(3). NE. ITMP(3)) GOTO 200                     ;DAY
        I1=IHR*60+IMIN
        CALL VALUE(IRRNM(4), 1, 1, X1, $200)
        CALL VALUE(IRRNM(5), 1, 1, X2, $200)
        X3=X1*60. 0+X2
        I2=INT(X3+0. 5)
        IF((I1-I2). GT. 20) GOTO 200
        IF((IGLNM. LT. 1). OR. (IGLNM. GT. 20)) GOTO 200
        WRITE(TTO, 8000) IRRNM
        GOTO 1000

C       FROM ELSEWHERE! GET FILE NAME AND SKIP TO SEND
C
200     WRITE(TTO, 9020)

400     ITMP(1)="FI"
        ITMP(2)="LE"
        ITMP(3)="NA"
        ITMP(4)="ME"
        ITMP(5)=". R"
        ITMP(6)="R<0>"
        CALL INPTS(IRRNM, ITMP, 9, $8040, $9020)
C       ILEN=LENG(IRRNM, 9)
C       IF(ILEN. EQ. 10) GOTO 420

410     WRITE(TTO) " <7>FILE UNAVAILABLE"
        GOTO 400

C420    IER=77K
        IER=77K
C       CALL MAPST(IEXTX, 1, 3, 2, IRRNM, 9, 11, $9000)
        CALL OPENR(IRRNM, MCH, IER, $500)
        CALL CLOSE(ICH, IER, $9000)
        GOTO 3000

500     CALL RESET
        GOTO 410

C       APPEND QC LIMITS ONLY IF FROM DMPP1
C
1000    IANS="N<0>"
        CALL INPTS(ITMP, IANS, 1, $8060, $3000)
        IF(IANS. EQ. "N<0>") GOTO 3000
        IF(IANS. EQ. "Y<0>") GOTO 1100

        WRITE(TTO) " <7>"
        GOTO 1000
```

```
1100    CALL OPEN(IRRNM, MCH, IER, $9000)
        CALL POSITION(MCH, 1, -1, -1, IER, $1140)
1140    I1=0
1150        I1=I1+1
            CALL NULST(IGCFN, 1, 10)
            DO 1160 I=1, 7
1160        IGCFN(I)=IGCLS(I, I1)
            ICHR=IGCLS(10, I1)
C           ICODE=NPOS(ICHR, 1, MCODE, 4, 1)
            DO 1162 I2=1, 8
                ICHK=NCHAR(MCODE, 4, I2)
                IF(ICHK. EQ. ICHR) GOTO 1164
1162        CONTINUE
            GOTO 1990
1164        ICODE=I2
            CALL OPENR(IGCFN, LCH, IER, $1180)
            WRITE(TTO, 8090) (IGCFN(I), I=1, 10), ICHR, ICODE
            READ(LCH, 8100, END=1180, ERR=1180) (ITMP(I), I=1, 7), NM, ND, NN,
     1          (ITXT(I), I=1, 16)
            CALL MAPST(ITXT, 18, 25, 16, ILBDT, 4, 1, $1990)
            GOTO 1190

1180        WRITE(TTO, 8120) IGCFN
            CALL CLOSE(LCH, IER, $1990)
            GOTO 1990

1190        DO 1900 IND=1, NM
                READ(LCH, 8140, END=1180, ERR=1180) INDG, (NAME(I), I=1, 35)
                READ(LCH, 8160, END=1180, ERR=1180) (ICAS(I), I=1, 6), IUSE,
     1          RMDL, XMDL, A, B, C, D, KEYD
                READ(LCH, 8180, END=1180, ERR=1180) RMEAN, SIGR, KEYR, XLSA2,
     1          SMEAN, SIGS, KEYS, XLCT, CMEAN, SIGC, KEYC
                IF(IND. NE. INDG) GOTO 1180

C               COMPRESS CAS NUMBER
                IRCAS(5)=ICAS(6)
                ICHAR=NCHAR(ICAS, 6, 10)
                CALL MOVBT(ICHAR, IRCAS, 6, 9, $1890)
                ICHAR=NCHAR(ICAS, 6, 9)
                CALL MOVBT(ICHAR, IRCAS, 6, 8, $1890)
                ICHAR=NCHAR(ICAS, 6, 7)
                CALL MOVBT(ICHAR, IRCAS, 6, 7, $1890)
                IRCAS(3)=ICAS(3)
                IRCAS(2)=ICAS(2)
                IRCAS(1)=ICAS(1)
                CALL NULST(ITMP, 1, 20;
                CALL PADST(ITMP, 6)

                IF(IUSE. EQ. 3) GOTO 1900
                GOTO(1500, 1200, 1400, 1890, 1600, 1800, 1890, 1890) ICODE
C                      U    D    F    X    C    R    N    S

C               DUPLICATES
C
1200            IF(IUSE. EQ. 2) GOTO 1900
                X1=A+D4
                X2=B+D4
                X3=C+D4
                X4=D+D4
                ICHR=(KEYD. AND. 177400K). OR. "<0> "
                IF(ICH. NE. "C ") ICHR="M "
                ITMP(1)="LD"
                ITMP(2)="UP"
                ITMP(5)=" A"
                WRITE(MCH, 8200) (ITMP(I), I=1, 6), IRCAS, X1, ILBDT,
     1          ICHR, (IGCFN(J), J=1, 5)
                ITMP(5)=" B"
                WRITE(MCH, 8200) (ITMP(I), I=1, 6), IRCAS, X2, ILBDT,
     1          ICHR, (IGCFN(J), J=1, 5)
                ITMP(5)=" C"
                WRITE(MCH, 8200) (ITMP(I), I=1, 6), IRCAS, X3, ILBDT,
     1          ICHR, (IGCFN(J), J=1, 5)
                ITMP(5)=" D"
                WRITE(MCH, 8200) (ITMP(I), I=1, 6), IRCAS, X4, ILBDT,
     1          ICHR, (IGCFN(J), J=1, 5)
                GOTO 1900
```

```
C               SPIKES
C
1400            IF(IUSE.EQ.2) GOTO 1900
                X1=SMEAN-2*SIGS+D3
                X2=SMEAN-SIGS+D3
                X3=SMEAN+D3
                X4=SMEAN+SICS+D3
                X5=SMEAN+2*SIGS+D3
                ICHR=(KEYS.AND.177400K).OR."<O> "
                ITMP(1)="LS"
                ITMP(2)="PK"
                GOTO 1840

C               SURROGATE SPIKES
C
1500            IF(IUSE.NE.2) GOTO 1900
                X1=SMEAN-2*SIGS+D3
                X2=SMEAN-SIGS+D3
                X3=SMEAN+D3
                X4=SMEAN+SICS+D3
                X5=SMEAN+2*SIGS+D3
                ICHR=(KEYS.AND.177400K).OR."<O> "
                ITMP(1)="LS"
                ITMP(2)="SP"
                GOTO 1840

C               CONTROL STANDARDS
C
1600            IF(XLCT.LE.D3) GOTO 1900
                X1=CMEAN-2*SIGC+D3
                X2=CMEAN-SIGC+D3
                X3=CMEAN+D3
                X4=CMEAN+SIGC+D3
                X5=CMEAN+2*SIGC+D3
                ICHR=(KEYC.AND.177400K).OR."<O> "
                ITMP(1)="LC"
                ITMP(2)="ST"
                GOTO 1840

C               REAGENT BLANKS
C
1800            X1=RMEAN-2*SIGR+D4
                X2=RMEAN-SIGR+D4
                X3=RMEAN+D4
                X4=RMEAN+SIGR+D4
                X5=RMEAN+2*SIGR+D4
                ICHR=(KEYR.AND.177400K).OR."<O> "
                ITMP(1)="LR"
                ITMP(2)="BL"

1840            ITMP(5)=" L"
                ITMP(6)="CL"
                IF(ICHR.NE."C ") ICHR="M "
                WRITE(MCH,8200) (ITMP(I),I=1,6),IRCAS,X1,ILBDT,ICHR,
     1          (IGCFN(J),J=1,5)
                ITMP(6)="WL"
                WRITE(MCH,8200) (ITMP(I),I=1,6),IRCAS,X2,ILBDT,ICHR,
     1          (IGCFN(J),J=1,5)
                ITMP(5)=" A"
                ITMP(6)="VE"
                WRITE(MCH,8200) (ITMP(I),I=1,6),IRCAS,X3,ILBDT,ICHR,
     1          (IGCFN(J),J=1,5)
                ITMP(5)=" U"
                ITMP(6)="WL"
                WRITE(MCH,8200) (ITMP(I),I=1,6),IRCAS,X4,ILBDT,ICHR,
     1          (IGCFN(J),J=1,5)
                ITMP(6)="CL"
                WRITE(MCH,8200) (ITMP(I),I=1,6),IRCAS,X5,ILBDT,ICHR,
     1          (IGCFN(J),J=1,5)
                GOTO 1900
```

```
1890            WRITE(TTO) " <7>ERROR!"
1900        CONTINUE

            CALL CLOSE(LCH, IER, $2000)
            GOTO 2000

1990        WRITE(TTO) " <7>MAJOR ERROR!"

2000    IF(I1.EG.IGLNM) GOTO 2020
        GOTO 1150

2020    CALL RESET
C       CALL CLOSE(MCH, IER, $9000)

C       OUTPUT OPTIONS
C
3000    WRITE(TTO, 8300)
        IOMD="T<0>"
        CALL INPTS(ITMP, IOMD, 1, $8320, $5000)
        IF(IOMD.EG."T<0>") GOTO 3100
        IF(IOMD.EG."N<0>") GOTO 3200
        IF(IOMD.EG."P<0>") GOTO 3800
        WRITE(TTO) " <7>WRONG MODE"
        GOTO 3000

C       OUTPUT RUN RESULTS TO MAG TAPE
C
3100    IFLFN(1)="MT"
        IFLFN(2)=":#"
        CALL NULST(IFLFN, 5, 10)
        WRITE(TTO, 8340)
        ITMP(1)="1<0>"
        CALL INPTS(IANS, ITMP, 1, $8360, $3000)
        IFLFN(3)=IANS
        KCHAN=DSK
        CALL OPENW(IFLFN, KCHAN, IER, $3140)
        CALL OPENR(IRRNM, MCH, IER, $3120)
        CALL FCOPY(MCH, KCHAN, $3120, IE, IER)
3120    CALL CLOSE(MCH, IE)
        CALL CLOSE(KCHAN, IEE)

        WRITE(TTO, 8380)

        GOTO 5000

3140    CALL CLOSE(KCHAN, IE)
        IF(IER.EG.22K) WRITE(TTO, 8400) IER
        IF(IER.EG.31K) WRITE(TTO, 8420) IER
        IF(IER.EG.37K) WRITE(TTO, 8440) IER
        IF(IER.EG.50K) WRITE(TTO, 8460) IER
        IF(IER.GT.50K.AND.IER.LE.61K) WRITE(TTO, 8480) IER
        GOTO 3000

C       OUTPUT RUN RESULTS TO NOVA VIA MICRO
C
3200    CALL NULST(ITMP, 1, 40)

        ISTRT(1)="ST"
        ISTRT(2)="/R"
        ISTRT(3)="T<0>"
        ISTRT(4)=0
        ISTOP(1)="<57>B"
        ISTOP(2)="OM"
        ISTOP(3)="B<0>"
        ISTOP(4)=0

        WRITE(TTO, 8560)
        ITMP(1)="CR"
        CALL INPTS(IANS, ITMP, 1, $8580, $3000)

        WRITE(TTO, 8620)

        WRITE(TTO, 8640) ISTRT
```

```
C3340   READ(ICH,8650,END=3420,ERR=3420) (ITXT(I),I=1,40)
C       DO 3360 J=1,40
C           I1=40-J+1
C           IF(ITXT(I1).NE."  ") GOTO 3380
C           ITXT(I1)=0
C3360   CONTINUE
C3380   K=I1*2
C       ICHR1=NCHAR(ITXT,40,K)
C       IF(ICHR1.NE.20000K) GOTO 3400
C       ITXT(I1)=ITXT(I1).AND.177400K
C       K=K-1
C
C3400   WRITE(TTO,8660) (ITXT(I),I=1,40)
C
C       GOTO 3340

        CALL OPENR(IRRNM,MCH,IER,$3500)
        CALL FCOPY(MCH,TTO,IER,$3420,IE,IER)
3420    CALL CLOSE(MCH,IE)

3500    WRITE(TTO,8680) ISTOP
        CALL CLOSE(MCH,IE)
        WRITE(TTO,8820)

        GOTO 5000

C       OUTPUT RUN RESULTS TO THE PDP 11/70 MICROPROCESSOR INTERFACE
C
3800    WRITE(TTO) " <7>PDP 11<57>70 TRANSFER NOT AVAILABLE."
        GOTO 3000

5000    IANS="N<0>"
        CALL INPTS(ITMP,IANS,1,$8900,$5300)
        IF(IANS.EQ."N<0>") GOTO 5300
        IF(IANS.EQ."Y<0>") GOTO 5100
        WRITE(TTO) " <7>"
        GOTO 5000

5100    KCHAN=LPT
        CALL OPLPT(KCHAN,$5200)
        CALL OPENR(IRRNM,MCH,IER,$5200)
        CALL FCOPY(MCH,KCHAN,$5200,IE,IER)
5200    CALL CLOSE(MCH,IE)
        CALL CLOSE(KCHAN,IEE)

5300    IANS="N<0>"
        CALL INPTS(ITMP,IANS,1,$8920,$5700)
        IF(IANS.EQ."N<0>") GOTO 5700
        IF(IANS.EQ."Y<0>") GOTO 5400
        WRITE(TTO) " <7>"
        GOTO 5300

5400    CALL DELETE(IRRNM,IER,$9000)

5700    CALL RESET
        CALL LOADIT("DMPC")
```

```
8000    FORMAT(" ",/," RUN RESULTS APPEND AND SEND MODULE (DMPP2)",/,
   1        " RUN RESULTS FILE NAME: ",9A2)
8020    FORMAT(" ",/," RUN RESULTS SEND MODULE (DMPP2)",/)
8040    FORMAT("  ENTER COMPLETE FILENAME [",9A2,"] > ",Z)
8060    FORMAT("  APPEND GC LIMITS TO RR FILE [",A2,"] > ",Z)
8090    FORMAT("  OPEN GC<57>MT: ",10A2,"   CODE: ",A2,"    ICODE: ",I6)
8100    FORMAT(7A2,3I4,16A2,1X)
8120    FORMAT(" <7>UNABLE TO PROCESS GC<57>METHOD TEMPLATE FILE ",10A2)
8140    FORMAT(I3,1X,35A2,2X)
8160    FORMAT(6A2,1X,I3,1X,2F7.2,1X,4F10.3,A1,1X)
8180    FORMAT(2F7.2,A1,2(F8.2,2F6.1,A1),1X)

8200    FORMAT(" 4 ",6A2," C",5A2,1X,E10.3,1X,4A2,1X,A2,11X,5A2,10X)

8300    FORMAT(" RUN RESULTS OUTPUT OPTIONS:",/,
   1        "  T - COPY TO TAPE",/,
   1        "  N - SEND TO NOVA VIA MICRO BUFFER",/,
   1        "  P - SEND TO PDP 11/70 VIA MICRO BUFFER")
8320    FORMAT("  OPTION (T, N, OR P) [",A2,"] > ",Z)
8340    FORMAT(/," <7>MOUNT MAG TAPE AND INITIALIZE.")
8360    FORMAT(" OUTPUT RR FILE TO MT:# [",A2,"] > ",Z)
8380    FORMAT(" <7>DISMOUNT TAPE AND RECORD RR FILES OUTPUT")
8400    FORMAT(" <7>FILE DOES NOT EXIST. ERROR: ",OI4)
8420    FORMAT(" <7>IMPROPER TRACK NUMBER. ERROR: ",OI4)
8440    FORMAT( " <7>TAPE UNIT IN USE. ERROR: ",OI4)
8460    FORMAT(" <7>TAPE UNIT NOT READY. ERROR: ",OI6)
8480    FORMAT(" <7>GENERAL TAPE ERROR: ",OI4)

C8540   FORMAT(" <7>FILE ACCESS ERROR: ",OI6)
8560    FORMAT(/," <7>SWITCH ON MICRO INTERFACE BOX")
8580    FORMAT("  PRESS RETURN WHEN READY [",A2,"] > ",Z)
8620    FORMAT("1")
8640    FORMAT(1X,4A2)
C8650   FORMAT(40A2)
C8660   FORMAT(1X,40A2)
9690    FORMAT(1X,4A2)
8820    FORMAT("1<7>TASK COMPLETE. TURN OFF INTERFACE NOW",/)
8900    FORMAT("  OUTPUT RUN RESULTS TO LINEPRINTER [",A2,"] > ",Z)
8920    FORMAT("  DELETE RUN RESULTS FILE FROM DISK [",A2,"] > ",Z)

9000    WRITE(TTO,9010) IER
9010    FORMAT(" <7>ERROR FATAL: ",OI4)
9020    CALL RESET
        CALL LOADIT("EXEC")

        END
        SUBROUTINE FLCHK(IFILE, MWRDS, INFRT, IERTN, IER)

C       THIS ROUTINE SCANS THE DISK DEVICES TO FIND IFILE.
C       IF D:IFILE IS FOUND, THE ROUTINE RETURNS NORMALLY
C       AND IFILE IS MODIFIED TO CONTAIN THE PROPER DEVICE
C       ATTRIBUTE IN SPECIFIER FORM.  IF THE DISK DEVICES ARE
C       EXHAUSTED BEFORE IFILE IS FOUND, IFILE IS RETURNED
C       UNCHANGED AND THE RETURN IS TO INFRT (NOT FOUND).
C       IF AN UNEXPECTED ERROR IS DETECTED, IFILE
C       IS RETURNED UNCHANGED AND THE RETURN IS TO IERTN.
C       IER WILL BE SET TO THE APPROPRIATE VALUE.

        COMMON JFLNM(20)
        INTEGER IFILE(MWRDS)
        DO 10 I=1,8
            JFLNM(I)=0
10      CONTINUE
        DO 100 I=1,MWRDS
            JFLNM(I+1)=IFILE(I)
100     CONTINUE
        JFLNM(1)="0:"
200     CALL RENAME(JFLNM, JFLNM, IER, $300)
300     IF(IER.LT.0) IER=-IER
        IF(IER.EQ.36K) GOTO 500          ;ALREADY EXISTS
        IF(IER.EQ.20K) GOTO 700          ;DEVICE NUMBER EXCEEDED
        IF(IER.NE.22K) GOTO 900          ;UNEXPECTED ERROR
        JFLNM(1)=JFLNM(1)+400K           ;INCREMENT DEVICE #
        GOTO 200

500     DO 600 I=1,MWRDS
            IFILE(I)=JFLNM(I)
600     CONTINUE
        RETURN

700     RETURN INFRT                             ;NOT FOUND
900     RETURN IERTN                             ;ERROR
        END
```

```
      SUBROUTINE GOBBL(NSTRNG,MWRDS,LENN)
C
C     THIS SUBROUTINE GOBBLES UP BLANKS AT THE BEGINNING
C     AND AT THE END OF STRING NSTRNG AND RETURNS THE
C     REMAINING STRING. ISPC IS A LEFT JUSTIFIED SPACE.
C     IT IS REPLACED WITH ICHRO WHICH IS A NULL. LENN IS
C     RETURNED AS THE LENGTH OF THE REMAINING STRING.


C
C     USES LENG, NCHAR, NULST, AND MOVST
C
      INTEGER NSTRNG(MWRDS)
      ICHRO=0
      LENN=0
      ISPC=20000K
      LENN=LENG(NSTRNG,MWRDS)
      IF(LENN.EQ.0) RETURN

      I2=LENN
      DO 100 I1=1,LENN
          ICHR1=NCHAR(NSTRNG,MWRDS,I1)
          IF(ICHR1.NE.ISPC) GOTO 300
100   CONTINUE

      LENN=0
      CALL NULST(NSTRNG,1,MWRDS)
      RETURN

300   ICHR1=NCHAR(NSTRNG,MWRDS,I2)
      IF(ICHR1.NE.ISPC) GOTO 400
      I2=I2-1
      GOTO 300

400   LENN=I2-I1+1
C
C     THIS MOVES A STRING INTO ITSELF
C
C     CALL MOVST(NSTRNG,I1,I2,MWRDS,NSTRNG,MWRDS,-1,$700)
C     RETURN

      I3=0
      DO 500 I=I1,I2
          I3=I3+1
          ICHR1=NCHAR(NSTRNG,MWRDS,I)
          CALL MOVBT(ICHR1,NSTRNG,MWRDS,I3,$700)
500   CONTINUE
      CALL NULST(NSTRNG,LENN+1,MWRDS)
      RETURN

700   WRITE(10) " <7>GOBBL FAILURE"
      RETURN
      END
      SUBROUTINE INPTI(NEWINT,LASTINT,IFORMAT,IEND)
C
C     INPTI PROMPTS THE USER ON THE CONSOLE USING
C     A FORMAT SUPPLIED BY THE CALLING PROGRAM AND TYPES
C     AN INTEGER NUMBER WHICH REPRESENTS THE DEFAULT RESPONSE.
C     INPTI THEN GETS AN INTEGER NUMBER FROM THE CONSOLE
C     IF ONE IS TYPED.  IF NO NUMBER IS FOUND, THE SUBROUTINE
C     USES THE DEFAULT NUMBER.  ERROR RETURN ON CTRL-D
C
C     ARGUMENTS ARE:
C             NEWINT  = SET EQUAL TO INPUT OR DEFAULT NUMBER
C             LASTINT = LAST NUMBER (DEFAULT ON NULL RESPONSE)
C             IFORMAT = $LABEL OF FORMAT STATEMENT IN CALLING PROGRAM
C             IEND    = $LABEL FOR RETURN WHEN CTRL-D STRUCK

      COMMON INPUT(10)

      WRITE(10,IFORMAT) LASTINT
      READ(10,10,END=20,ERR=20) INPUT
10    FORMAT(10A1)
      IFLG=0
      ISGN=1
      NEWINT=0
      K=0
      DO 15 I=1,10
      J=INPUT(I)
      IF(J.EQ."  ") GO TO 15
      IF(IFLG.EQ.1) GO TO 11
      IF(J.NE."- ".AND.J.NE."+ ") GO TO 11
      IFLG=1
      IF(J.EQ."- ") ISGN=-ISGN
      GO TO 15
11    IF(J.LT."0 ".OR.J.GT."9 ") GOTO 18
      K=J/256
      NEWINT=10*NEWINT+K-60K
15    CONTINUE
18    NEWINT=ISIGN(NEWINT,ISGN)
      IF(K.EQ.0) NEWINT=LASTINT
      LASTINT=NEWINT
      RETURN
20    RETURN IEND
      END
```

```
        SUBROUTINE INPTF(XNEWVL, OLDVAL, IFORMAT, IEND)
C
C       INPTF PROMPTS THE USER ON THE CONSOLE USING
C       A FORMAT SUPPLIED BY THE CALLING PROGRAM AND TYPES
C       A FLOATING NUMBER WHICH REPRESENTS THE DEFAULT RESPONSE.
C       INPTF THEN GETS A FLOATING NUMBER FROM THE CONSOLE.
C       IF NO NUMBER IS FOUND, THE SUBROUTINE USES THE DEFAULT RESPONSE.
C       INPTF RETURNS TO THE LOCATION "IEND" IF EOF IS ENCOUNTERED.
C
C       ARGUMENTS ARE:
C               XNEWVL  = SET EQUAL TO INPUT OR DEFAULT NUMBER.
C               OLDVAL  = LAST NUMBER (DEFAULT VALUE)
C               IFORMAT = $LABEL OF FORMAT STATEMENT IN CALLING PROGRAM
C               IEND    = $LABEL FOR RETURN WHEN CTRL-D STRUCK

        COMMON INPUT(15)

        T=OLDVAL+SIGN(0.0005, OLDVAL)
        WRITE(10, IFORMAT) T
        READ(10, 100, END=200, ERR=200) INPUT
100     FORMAT(15A1)

        XNEWVL=0.
        IFLG=0
        ISGN=1
        K=0
        DO 15 I=1, 15
        J=INPUT(I)
        IF(J.EQ."  ") GOTO 15
        IF(IFLG.EQ.1) GOTO 11
        IF(J.NE."- ".AND.J.NE."+ ") GOTO 11
        IFLG=1
        IF(J.EQ."- ") ISGN=-ISGN
        GOTO 15
11      IF(J.NE.". ") GOTO 12
        IDEC=1
        GOTO 10
12      IF(J.LT."0 ".OR.J.GT."9 ") GOTO 5
        K=J/256
        XNEWVL=10.*XNEWVL+FLOAT(K-60K)
15      CONTINUE
        GOTO 5

10      DEC=1.
        I=I+1
        DO 20 I=I, 15
        J=INPUT(I)
        IF(J.LT."0 ".OR.J.GT."9 ") GOTO 5
        DEC=DEC*10.
        K=J/256
        XNEWVL=XNEWVL+FLOAT(K-60K)/DEC
20      CONTINUE

5       XNEWVL=SIGN(XNEWVL, FLOAT(ISGN))
        IF(K.EQ.0) XNEWVL=OLDVAL
        OLDVAL=XNEWVL
        RETURN
200     RETURN IEND
        END
```

```
      SUBROUTINE ,INPTS(NEWSTR,LASTSTR,LENGTH,IFORMAT,IEND)

C        INPTS PROMPTS THE USER ON THE CONSOLE DEVICE USING A FORMAT
C        SUPPLIED BY THE CALLING PROGRAM AND TYPES AN ALPHANUMERIC
C        STRING WHICH REPRESENTS THE DEFAULT ANSWER TO THE PROMPT.
C        THE SUBROUTINE THEN GETS A CHARACTER STRING FROM THE CONSOLE AND,
C        IF IT NOT BLANK, STORES IT IN THE ARRAY IN THE CALLING PROGRAM.
C        IF THE STRING IS NULL OR ALL BLANKS, THE DEFAULT ANSWER IS USED.
C
C        ARGUMENTS ARE:
C                NEWSTR  = ARRAY FOR ANSWER
C                LASTSTR = PREVIOUS ANSWER
C                LENGTH  = NUMBER OF WORDS IN ARRAY
C                IFORMAT = $LABEL OF FORMAT STATEMENT IN CALLING PROGRAM
C                IEND    = $LABEL FOR RETURN WHEN CTRL-D HIT.
C
C        INPTS USES FUNCTION NCHAR.
C

         COMMON INPUT(40)
         INTEGER NEWSTR(LENGTH),LASTSTR(LENGTH)

         WRITE(10,IFORMAT) (LASTSTR(I),I=1,LENGTH)
         READ(10,10,END=20) INPUT
10       FORMAT(40A2)
C
C        LIN=LENG(INPUT,LENGTH)
C        WRITE(10,1000) (INPUT(I),I=1,40),LIN
C1000    FORMAT(" INPUT = ",40A2,"     WITH LENGTH = ",I3)
C
         DO 1 J=1,40
         I=40-J+1
         IF(INPUT(I).NE."  ") GO TO 3
1        INPUT(I)=0
         DO 2 I=1,LENGTH
2        NEWSTR(I)=LASTSTR(I)
         RETURN
3        K=I*2
         ICHR1=NCHAR(INPUT,LENGTH,K)
         IF(ICHR1.NE.20000K) GO TO 4
         INPUT(I)=INPUT(I).AND.177400K
4        DO 5 I=1,LENGTH
         LASTSTR(I)=INPUT(I)
5        NEWSTR(I)=INPUT(I)
         RETURN
20       RETURN IEND
         END
      FUNCTION LENG(NSTRNG,MWRDS)
C
C     THIS FUNCTION RETURNS THE LENGTH OF NRSTRNG.  CHARACTERS
C     ARE EXAMINED UNTIL THE FIRST NULL IS ENCOUNTERED.
C     CHARACTERS FOLLOWING THE FIRST NULL ARE DISREGARDED.
C
C     USES FUNCTION ISHFT,NCHAR
C
      INTEGER NSTRNG(MWRDS)

      LENG=0
      IBTLN=2*MWRDS

      I=0
100   I=I+1
      IWORD=(I+1)/2
      IMD=(1-MOD(I,2))
      ICHR=NSTRNG(IWORD)
      I1=ISHFT(ICHR,8*IMD).AND.177400K
      IF(I1.EQ.0) GOTO 400
      LENG=LENG+1
      IF(I.EQ.IBTLN) GOTO 400
      GOTO 100

400   RETURN
      END
      SUBROUTINE MAPST(INSTR, ISTRT, ISTOP, NWRDS, NSTRNG, MWRDS, NBYTE, IERTN)
C
C     THIS SUBROUTINE MAPS STRING INSTR ONTO STRING NSTRNG AT
C     POSITION NBYTE.  THIS ROUTINE WILL WRITE IN NULL SPACE.
C
C     USES NCHAR, LENG, AND MOVBT
C
      INTEGER INSTR(NWRDS), NSTRNG(MWRDS)
      IF(NBYTE. LE. 0) GOTO 700
      NBYT=NBYTE
      LENI=LENG(INSTR, NWRDS)
      IF (LENI. LE. 0) RETURN
      IF (ISTRT. LT. 1. OR. ISTRT. GT. LENI) GOTO 700
      IF (ISTOP. GT. LENI) ISTOP=LENI
      IF (ISTOP. LT. ISTRT) GOTO 700
      LENN=LENG(NSTRNG, MWRDS)
      IF (NBYT. GT. (MWRDS*2)) GOTO 700
C
C     WRITE INSTR INTO OR ONTO NSTR
C
      DO 300 I=ISTRT, ISTOP
              ICH=NCHAR(INSTR, NWRDS, I)
              IBT=NBYT+I-ISTRT
              CALL MOVBT(ICH, NSTRNG, MWRDS, IBT, $700)
300   CONTINUE
      RETURN

700   RETURN IERTN
      END
      SUBROUTINE MOVBT(ICHAR,NSTRNG,MWRDS,IBYTE,IERTN)
C
C     THIS SUBROUTINE INSERTS CHARACTER ICHAR IN STRING NSTRNG
C     AT POSITION IBYTE.  ICHAR IS EXPECTED TO BE LEFT JUSTIFIED
C     IN THE FIELD.  IERTN IS AN ERROR RETURN.
C
C     USES FUNCTION ISHFT AND MOD
C
      INTEGER NSTRNG(MWRDS)
      IF(IBYTE.LE.0) GOTO 700
      IWORD=(IBYTE+1)/2
      IF(IWORD.GT.MWRDS) GOTO 700
      II=MOD(IBYTE,2)

      ICHR1=(NSTRNG(IWORD).AND.ISHFT(177400K,-8*II))
      ICHR2=(ISHFT(ICHAR,8*(II-1)).AND.ISHFT(177400K,8*(II-1)))
      NSTRNG(IWORD)=ICHR1.OR.ICHR2

      RETURN

700   RETURN IERTN
      END
      SUBROUTINE MOVST(INSTR, ISTRT, ISTOP, NWRDS, NSTRNG, MWRDS, NBYTE, IERTN)
C
C     THIS SUBROUTINE MOVES STRING INSTR INTO STRING NSTRNG AT
C     POSITION NBYTE.  THREE VARIATIONS ARE POSIBLE DEPENDING
C     ON THE VALUE OF NBYTE:
C           NBYTE > 0  INSERT INSTR INTO NSTRNG AT NBYTE
C           NBYTE = 0  APPEND INSTR ONTO NSTRNG
C           NBYTE < 0  OVERWRITE INSTR INTO NSTRNG AT NBYTE
C
C     NBYTE IS RETURNED AS THE LENGTH OF THE RESULTING STRING
C     USES NCHAR, LENG, AND MOVBT
C
      INTEGER INSTR(NWRDS),NSTRNG(MWRDS)
      ICHRO=0
      NBYT=NBYTE
      LENI=LENG(INSTR,NWRDS)
      IF (LENI.LE.0) RETURN
      IF (ISTRT.LT.1.OR.ISTRT.GT.LENI) GOTO 700
      IF (ISTOP.GT.LENI) ISTOP=LENI
      IF (ISTOP.LT.ISTRT) GOTO 700
      LENN=LENG(NSTRNG,MWRDS)
      IF (NBYT.LT.0) NBYT=-NBYT
      IF (NBYT.GT.(LENN+1)) GOTO 700

C     ROUTE FOR OVERWRITE, APPEND, OR INSERT
      IF (NBYTE) 200,150,100
100   IF (LENN.EQ.0) GO TO 150

C     SPREAD SEGMENTS OF NSTRNG TO ALLOW INSERTION OF INSTR
      I=LENN
110   ICH=NCHAR(NSTRNG,MWRDS,I)
      IBT=I+ISTOP-ISTRT+1
      CALL MOVBT(ICH,NSTRNG,MWRDS,IBT,$700)
      I=I-1
      IF (I.GE.NBYT) GO TO 110
      GO TO 200

C     APPEND INSTR TO NSTRNG AT NBYT
150   NBYT=LENN+1

C     WRITE INSTR INTO OR ONTO NSTR
200   DO 300 I=ISTRT,ISTOP
            ICH=NCHAR(INSTR,NWRDS,I)
            IBT=NBYT+I-ISTRT
            CALL MOVBT(ICH,NSTRNG,MWRDS,IBT,$700)
300   CONTINUE
      IF(NBYTE.LT.0) GOTO 400
      NBYTE=NBYT+ISTOP-ISTRT
      RETURN

C     CLEANUP AFTER OVERWRITE
400   NBYTE=IBT
      IF(IBT.GE.LENN) RETURN
      IBT=IBT+1
      DO 500 I=IBT,LENN
            CALL MOVBT(ICHRO,NSTRNG,MWRDS,I,$700)
500   CONTINUE
      RETURN

700   RETURN IERTN
      END
      SUBROUTINE NCAS(ISTRN, MWRDS, IERTN)

C     THIS FUNCTION GOBBLES THE SPACES AROUND STRING ISTRN
C     AND THEN CHECKS THE CAS NUMBER TO DETERMINE
C     IF IT IS LEGITIMATE.  IERTN IS TAKEN FOR INVALID CAS CODES.

C     NCAS USES NCHAR AND GOBBL.

      INTEGER ISTRN(MWRDS)
      IF(MWRDS.NE.6) GOTO 700
      CALL GOBBL(ISTRN, MWRDS, LCAS)
      IF((LCAS.LT.4).OR.(LCAS.GT.11)) GOTO 700

      ICHR=NCHAR(ISTRN, MWRDS, LCAS)
      ICHK0=ICHR/256-48
      IF((ICHK0.LT.0).OR.(ICHK0.GT.9)) GOTO 700
      ISTRN(6)=ISHFT(ICHR,-8).OR."-<0>"

      ISUM=0
      IPOS=11
      I=1
200   LCAS=LCAS-1
      IF(LCAS.LE.0) GOTO 400
      ICH=NCHAR(ISTRN, MWRDS, LCAS)
      IF((ICH.EQ." <0>").OR.(ICH.EQ."-<0>")) GOTO 200
      IDIG=ICH/256-48
      IF((IDIG.LT.0).OR.(IDIG.GT.9)) GOTO 700
      ISUM=ISUM+IDIG*I
      I=I+1
300   IPOS=IPOS-1
      CALL MOVBT(ICH, ISTRN, MWRDS, IPOS, $700)
      IF(IPOS.NE.9) GOTO 200
      ICH="-<0>"
      GOTO 300

400   ICH=" <0>"
      IPOS=IPOS-1
      DO 500 J=1, IPOS
         CALL MOVBT(ICH, ISTRN, MWRDS, J, $700)
500   CONTINUE

      IF(IPOS.GT.6) GOTO 700
      ICHK1=MOD(ISUM, 10)
      IF(ICHK0.NE.ICHK1) GOTO 700
      RETURN

700   RETURN IERTN
      END
      FUNCTION NCHAR(NSTRNG,MWRDS,IBYTE)
C
C     THIS FUNCTION RETURNS THE CHARACTER IN POSITION IBYTE OF
C     NSTRNG.  THE CHARACTER IS LEFT JUSTIFIED IN THE WORD NCHAR.
C     A NULL IS RETURNED IF ERRORS ARE ENCOUNTERED.
C
C     USES FUNCTION ISHFT
C

      INTEGER NSTRNG(MWRDS)
      NCHAR=0K
      IF(IBYTE.LE.0) RETURN
      IWORD=(IBYTE+1)/2
      IF(IWORD.GT.MWRDS) RETURN

      II=ISHFT(NSTRNG(IWORD),8*(1-MOD(IBYTE,2))).AND.177400K
      NCHAR=II

      RETURN
      END
      FUNCTION NPOS(INSTR, NWRDS, NSTRNG, MWRDS, IBYTE)
C
C     THIS FUNCTION RETURNS THE STARTING POSITION OF THE OCCURRENCE
C     OF INSTR IN NSTRNG STARTING WITH POSITON IBYTE IN NSTRNG. THE
C     FUNCTION RETURNS THE FOLLOWING VALUES:
C
C     NPOS =  0  IF INSTR NOT FOUND IN NSTRNG.
C     NPOS >  0  INDICATING THE BYTE POSITION OF THE START OF
C                INSTR IN NSTRNG.
C
C     USES LENG, ISHFT, AND MOD
C
      INTEGER NSTRNG(MWRDS), INSTR(NWRDS)
      NPOS=0
      LENI=LENG(INSTR, NWRDS)
      IF(LENI. EG. 0; RETURN
      LENB=2*MWRDS
      IF(IBYTE. LE. 1) IPOS=0
      IF(IBYTE. GT. 1) IPOS=IBYTE-1
100   ILEFT=LENB-IPOS
      IF (LENI. GT. ILEFT) RETURN
C
C     FIND OUT IF INSTR IS IN NSTRNG STARTING AT IPOS
C
      DO 200 I=1, LENI
          IX=IPOS+I
          IWORD=(IX+1)/2
          IMD=MOD(IX, 2)
          ICHR1=ISHFT(NSTRNG(IWORD), -8*IMD). AND. 377K
          IWORD=(I+1)/2
          IMD=MOD(7, 2)
          ICHR2=ISHFT(INSTR(IWORD), -8*IMD). AND. 377K
          IF(ICHP1. EQ. ICHR2) GOTO 200
          IPOS=IPOS+1
          GOTO 100
200   CONTINUE
C
C     SUCCESS - STRING FOUND
C
      NPOS=IPOS+1
      RETURN
      END
```

```
         SUBROUTINE NULST(NSTRNG, ISTBT, MWRDS)

C        THIS SUBROUTINE WILL NULL A STRING STARTING FROM BYTE ISTBT
C        AND EXTENDING TO THE END OF THE STRING (IE: MWRDS*2 BYTES).
C        IF ISTBT IS LESS THAN OR EQUAL TO ZERO, THE WHOLE STRING IS
C        MADE  A NULL STRING. IF THE START BYTE IS OUT OF THE RANGE
C        OF THE WORD, A RETURN IS MADE WITH NO ACTION.

         INTEGER NSTRNG(MWRDS)

         IF(ISTBT.LE.0) ISTBT=1
         IWDST=(ISTBT+1)/2
         IF(IWDST.GT.MWRDS) RETURN
         II=MOD(ISTBT,2)
         IF(II.EQ.0) NSTRNG(IWDST)=NSTRNG(IWDST).AND.177400K
         IF(II.EQ.1) NSTRNG(IWDST)=0
         IF(IWDST.EQ.MWRDS) RETURN

         I1=IWDST+1
         DO 100 I=I1,MWRDS
                NSTRNG(I)=0
100      CONTINUE

         RETURN
         END
```

```
      SUBROUTINE PADST(NSTRNG,MWRDS)

C     THIS SUBROUTINE WILL PAD A STRING WITH BLANKS
C     FROM THE FIRST NULL UP TO THE MWRDS*2 BYTE.

C     USES MOD. AND LENG SUBROUTINES.

      INTEGER NSTRNG(MWRDS)

      IBYT=LENG(NSTRNG,MWRDS)+1
      IWRD=(IBYT+1)/2
      IF(IWRD.GT.MWRDS) RETURN
      II=MOD(IBYT,2)
      IF(II.EQ.0) NSTRNG(IWRD)=NSTRNG(IWRD).OR.40K
      IF(II.EQ.1) NSTRNG(IWRD)=20040K
      IF(IWRD.EQ.MWRDS) RETURN

      I1=IWRD+1
      DO 100 I=I1,MWRDS
              NSTRNG(I)=20040K
100   CONTINUE

      RETURN
      END
      SUBROUTINE QCTST(XCONC, XMEAN, XSIG, XKNWN, IQMSG, PREC)
C
C     QCTST COMPUTES A PERCENT RECOVERY WITH XCONC AND XKNWN
C     FOR SPIKES AND CONTROL STANDARDS.  IF XKNWN.LE.0.0 THEN
C     THE ROUTINE ASSUMES A REAGENT BLANK WAS RUN AND PREC IS
C     SET TO XCONC.  IF APPLICABLE, QCTST THEN COMPARES THE PREC
C     WITH THE CONTROL LIMITS COMPUTED FROM XMEAN AND
C     XSIG.  IQMSG IS ASSIGNED THE VALUE OF "NO"
C     IF THE PREC IS GREATER THAN 3*XSIG FROM XMEAN;
C     "WR" IF PREC IS BETWEEN 2 AND 3*XSIG FROM
C     XMEAN; AND "OK" IF PREC IS WITHIN 2*XSIG
C     OF XMEAN.
C
C     USES FUNCTION ABS

      IQMSG="NO"
      IF(XKNWN.GT.0.0001) GOTO 100
      PREC=XCONC
      GOTO 200
100   PREC=100.0*XCONC/XKNWN
200   TEST=ABS(PREC-XMEAN)
      IF(TEST.LE.(3.0*XSIG)) IQMSG="WR"
      IF(TEST.LE.(2.0*XSIG)) IQMSG="OK"
      RETURN
      END
```

```
        SUBROUTINE VALUE(NSTRNG,MWRDS,ISTRT,VAL,IERTN)
C
C       THIS FUNCTION TAKES NSTRNG AND CONVERTS IT TO A FLOATING
C       POINT NUMBER. THE ROUTINE FINDS THE FIRST POSSIBLE NUMBER.
C       ANY TRAILING NON-NUMERIC CHARACTERS TRIGGER AN EXIT.  AN
C       IERTN IS TAKEN IF NO DIGITS ARE ENCOUNTERED.
C
C       USES FUNCTION NCHAR
C
        INTEGER NSTRNG(MWRDS)
        ISTOP=MWRDS*2
        VAL=0.0
        IF(ISTRT.GT.ISTOP) GOTO 700
        IFLG=0
        ISGN=1
        K=0

        DO 300 I=ISTRT,ISTOP
            J=NCHAR(NSTRNG,MWRDS,I)
            IF(J.EQ." <0>") GOTO 300
            IF(IFLG.EQ.1) GOTO 100
            IF(J.NE."-<0>".AND.J.NE."+<0>") GOTO 100
            IFLG=1
            IF(J.EQ."-<0>") ISGN=-ISGN
            GOTO 300
100         IF(J.NE.".<0>") GOTO 200
            IDEC=1
            GOTO 400
200         IF(J.LT."0<0>".OR.J.GT."9<0>") GOTO 600
            K=J/256
            VAL=10.0*VAL+FLOAT(K-60K)
300     CONTINUE
        GOTO 600

400     DEC=1.0
        I=I+1
        DO 500 I1=I,ISTOP
            J=NCHAR(NSTRNG,MWRDS,I1)
            IF(J.LT."0<0>".OR.J.GT."9<0>") GOTO 600
            DEC=DEC*10.0
            K=J/256
            VAL=VAL+FLOAT(K-60K)/DEC
500     CONTINUE

600     IF(K.EQ.0) GOTO 700
        VAL=SIGN(VAL,FLOAT(ISGN))
        RETURN

700     RETURN IERTN
        END
```